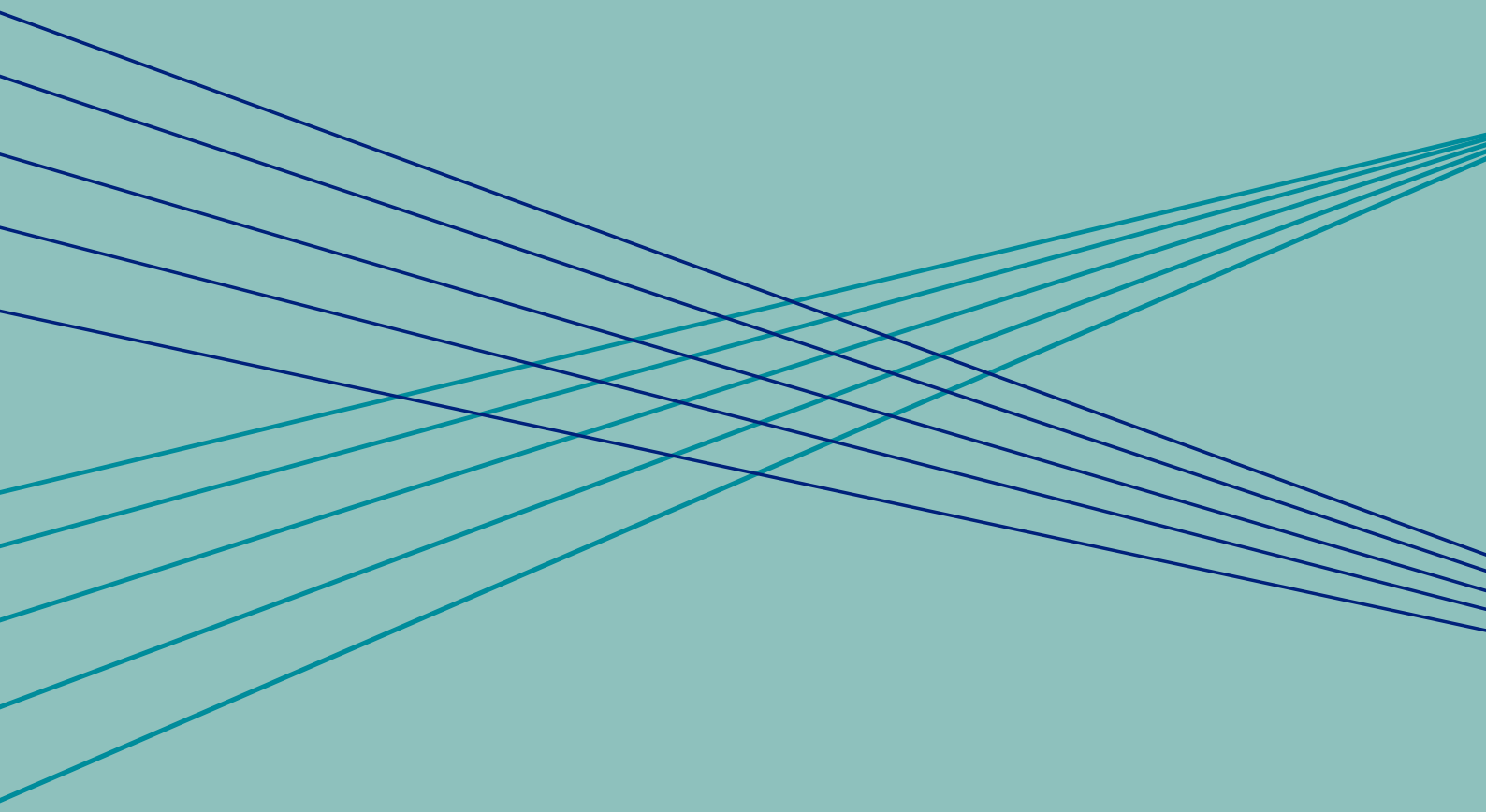
Integrating Physics Modelling with Machine Learning for Remote Sensing

Autor: Daniel Heestermans Svendsen
Directors: Gustau Camps-Valls i Luca Martino



Doctorat en Enginyeria Electrònica
Juliol de 2020

VNIVERSITAT DE VALÈNCIA

TESI DOCTORAL EN ENGINYERIA ELECTRÒNICA

PER

DANIEL HEESTERMANS SVENDSEN

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA

VNIVERSITAT DE VALÈNCIA

INTEGRATING PHYSICS MODELLING WITH MACHINE LEARNING FOR REMOTE SENSING

Directors:

GUSTAU CAMPS VALLS

LUCA MARTINO

DR. GUSTAU CAMPS VALLS, Doctor en Física, Catedràtic d'Universitat al Departament d'Enginyeria Electrònica de l'Escola Tècnica Superior D'Enginyeria de la Universitat de València

DR. LUCA MARTINO, Doctor en Processament del Senyal, Professor (acreditat a Titular) d'Universitat de Universidad Rey Juan Carlos (Departament de Teoria del Senyal)

FAN CONSTAR QUE:

DANIEL HEESTERMANS SVENDSEN, Bsc in Physics and Nanotechnology i MSc in Mathematical Modelling and Computation, ha realitzat sota la seva direcció el treball titulat *Integrating physics modelling with machine learning for remote sensing*, que es presenta en aquesta memòria per optar al grau de Doctor per la Universitat de València.

I per tal què així conste a efectes oportuns, i donant el vistiplau per a la presentació d'aquest treball davant el Tribunal de tesi que corresponga, signem el present certificat a València el 7 de Juliol 2020.

Gustau Camps Valls

Luca Martino

TESI DOCTORAL:

Integrating physics modelling with machine learning for remote sensing

AUTOR:

Daniel Heestermans Svendsen

DIRECTORS:

Dr. Gustau Camps Valls
Dr. Luca Martino

El tribunal nombrat per jutjar la Tesi Doctoral citada anteriorment, compost per:

President: _____

Vocal: _____

Secretari: _____

Acorda otorgar-li la qualificació de _____

I per a què així conste a efectes oportuns, signem el present certificat.

A Paterna el de de 2020

NOTE TO THE READER

According to the University of Valencia Doctorate Regulation[1] this PhD thesis is presented as a compendium of at least three publications in international journals containing the results of the conducted work. This thesis describes the published methods and the context within which they were developed. It also describes work that has recently been submitted to scientific journals. Furthermore, in accordance with the aforementioned regulation, and with the aim to foster the language of the University of Valencia in research and educational activity, the thesis also includes an extended abstract in Valencian.

[1] Reglament sobre depòsit, avaluació i defensa de la tesi doctoral aprovat pel Consell de Govern de 28 de Juny de 2016. ACGUV 172/2016.
Pla d'increment de la docència en valencià (ACGUV 129/2012) aprovat i modificat pel Consell de Govern de 22 de desembre de 2016. ACGUV 308/2016.

ACKNOWLEDGEMENTS

Carrying out a large project spanning over several years such as a PhD thesis, usually requires more than sheer determination and sweat of the brow. Personally, I have received a great deal of support, direct and indirect, from collaborators, friends and family.

I am lucky enough to be able to call the people who supervised my thesis, Gustau Camps-Valls and Luca Martino, my friends. They have, from the beginning, made me feel that they are on my side and willing to help with matters both academic and otherwise. This has made for a truly pleasant work environment. I am grateful to Gustau for the opportunities offered to me through this PhD position and for maintaining a creative environment in which I could play a large part in shaping the direction of the research we did. Furthermore, the extensive research carried out by Luca and I over the years on the difference between Sicilian and Danish personal space has been a hilarious and truly enriching one.

I consider myself lucky to have made such a cool and thoughtful friend as Emmanuel Johnson. Our friendship, and all the experiences (and countless tostadas) that we have shared have meant the world to me during my time as a PhD student. I am also thankful for my lovely family and Eskil Ferslev, Ida Aamot and Julia Roig for always being there for me.

The Image and Signal Processing group in its entirety is a very warm place to be, literally and figuratively (I had to include a bad heat-joke). The fun adventures shared during summer schools, conferences and road trips with Gonzalo, Emiliano and Diego have made my experience at the ISP a great one. So has the many beach trips, bike excursions and nights out shared with David, Anna, Jordi C, Nieves, Shari, Eatidal, Roberto, Charlotte, Laura, Dan, Adrian and Jose. I am thankful to Valero for the life advice, the ping-pong, the darts, and I guess being updated on the newest developments in machine learning is not a bad thing either. I owe thanks to Jordi, for his welcoming attitude and willingness to share what he knows about Unix and remote computing, which is everything. I am also grateful to Ana, Maria, Alvaro, Jesus, José, Luis and Julia for all the talks, both the ones given in front of the whiteboard and the ones had over the lunch-table in the cafeteria.

I owe special thanks to Daniel Hernández-Lobato for inviting me to have my research stay in the Machine Learning group at the Universidad Autónoma de Madrid, and for answering so many questions about neural networks and variational inference. I am also grateful to for Rafael Molina welcoming me into his group for a short stay while the great Pablo Morales-Álvarez explained to me what a deep GP was and showed me around the beautiful city of Granada.

Contents

List of Figures

List of Tables

Acronyms

ADAM	ADAptive Moment estimation optimization algorithm
AL	Active Learning
AMOGAPE	Active Multi-Output Gaussian Process Emulator
CDOM	Coloured Dissolved Organic Matter
Chl	Chlorophyll content
Cm	Dry matter content
Cw	Water content
DGP	Deep Gaussian Process
ELBO	Evidence Lower BOund
EM	Expectation Maximiztion
EQ	Exponentiated Quadratic
FITC	Fully Independent Training Conditional
GP	Gaussian Process
HMC	Hamiltonian Monte Carlo
IASI	Infrared Atmospheric Sounding Interferometer
KRR	Kernel Ridge Regression
KLD	Kullback-Leibler Divergence
LAI	Leaf Area Index
LANDSAT	LAND observation SATellite

LFM	Latent Force Model
LHS	Latin Hypercube Sampling
LOO	Leave-One-Out
LUT	Look-Up Table
MCEM	Monte Carlo Expectation Maximiztion
MCMC	Monte Carlo Markov Chain
MOGP	Multi-Output Gaussian Process
NDVI	Normalized Difference Vegetation Index
NN	Neural Network
RMSE	Root Mean Square Error
RTM	Radiative Transfer Model
SAIL	Scattering by Arbitrary Inclined Leaves
SM	Soil Moisture
SVGP	Scalable Variational Gaussian Process
TOA	Top of Atmosphere
TOC	Top of Canopy
ODE	Ordinary Differential Equation
OLI	Operational Land Imager
RFR	Random Forest Regression
RS	Remote Sensing
VAE	Variational Auto-Encoder
VI	Variational Inference

Abstract

Earth observation through satellite sensors, models and in situ measurements provides a way to monitor our planet with unprecedented spatial and temporal resolution. The amount and diversity of the data which is recorded and made available is ever-increasing. This data allows us to perform crop yield prediction, track land-use change such as deforestation, monitor and respond to natural disasters and predict and mitigate climate change.

The last two decades have seen a large increase in the application of machine learning algorithms in Earth observation in order to make efficient use of the growing data-stream. Machine learning algorithms, however, are typically model agnostic and too flexible and so end up not respecting fundamental laws of physics. On the other hand there has, in recent years, been an increase in research attempting to embed physics knowledge in machine learning algorithms in order to obtain interpretable and physically meaningful solutions. The main objective of this thesis is to explore different ways of encoding physical knowledge to provide machine learning methods tailored for specific problems in remote sensing.

Ways of expressing expert knowledge about the relevant physical systems in remote sensing abound, ranging from simple relations between reflectance indices and biophysical parameters to complex models that compute the radiative transfer of electromagnetic radiation through our atmosphere, and differential equations that explain the dynamics of key parameters.

This thesis focuses on inversion problems, emulation of radiative transfer models, and incorporation of the above-mentioned domain knowledge in machine learning algorithms for remote sensing applications. We explore new methods that can optimally model simulated and in-situ data jointly, incorporate differential equations in machine learning algorithms, handle more complex inversion problems and large-scale data, obtain accurate and computationally efficient emulators that are consistent with physical models, and that efficiently perform approximate Bayesian inversion over radiative transfer models.

RESUM

L'observació de la Terra a partir de les dades proporcionades per sensors abord de satèl·lits, així com les proporcionades per models de transferència radiativa o climàtics, juntament amb les mesures in situ proporcionen una manera sense precedents de monitorar el nostre planeta amb millors resolucions espacials i temporals. La riquesa, quantitat i diversitat de les dades adquirides i posades a disposició també augmenta molt ràpidament. Aquestes dades ens permeten predir el rendiment dels cultius, fer un seguiment del canvi d'ús del sòl com ara la desforestació, supervisar i respondre als desastres naturals, i predir i mitigar el canvi climàtic.

Per tal de fer front a tots aquests reptes, les dues darreres dècades han evidenciat un gran augment en l'aplicació d'algorismes d'aprenentatge automàtic en l'observació de la Terra. Amb l'anomenat 'machine learning' es pot fer un ús eficient del flux de dades creixent en quantitat i diversitat. Els algorismes d'aprenentatge màquina, però, solen ser models agnòstics i massa flexibles i, per tant, acaben per no respectar les lleis fonamentals de la física. D'altra banda, en els darrers anys s'ha produït un augment de la investigació que intenta integrar el coneixement de física en algorismes d'aprenentatge, amb la finalitat d'obtenir solucions interpretables i que tinguin sentit físic.

L'objectiu principal d'aquesta tesi és dissenyar diferents maneres de codificar el coneixement físic per proporcionar mètodes d'aprenentatge automàtic adaptats a problemes específics en teledetecció. Introduïm nous mètodes que poden fusionar de manera òptima fonts de dades heterogènies, explotar les regularitats de dades, incorporar equacions diferencials, obtenir models precisos que emulen, i per tant són coherents amb models físics, i models que aprenen parametrizacions del sistema combinant models i simulacions.

RESUMEN

La observación de la Tierra a partir de los datos proporcionados por sensores abordo de satélites, por modelos de transferencia radiativa o climáticos, junto con las medidas in situ proporcionan una manera sin precedentes de monitorizar nuestro planeta con mejores resoluciones espaciales y temporales. La riqueza, cantidad y diversidad de los datos adquiridos y puestos a disposición también aumenta muy rápidamente. Estos datos nos permiten predecir el rendimiento de los cultivos, hacer un seguimiento del cambio de uso del suelo como la deforestación, supervisar y responder a los desastres naturales, y predecir y mitigar el cambio climático.

Con el fin de hacer frente a todos estos retos, las dos últimas décadas han evidenciado un gran aumento en la aplicación de algoritmos de aprendizaje automático en la observación de la Tierra. Con el llamado 'machine learning' se puede hacer un uso eficiente del flujo de datos que crece constantemente en cantidad y diversidad. Los algoritmos de aprendizaje máquina, sin embargo, suelen ser modelos agnósticos y demasiado flexibles y, por tanto, acaban por no respetar las leyes fundamentales de la física. Por otra parte, en los últimos años se ha producido un aumento de la investigación que intenta integrar el conocimiento de física en algoritmos de aprendizaje, con el fin de obtener soluciones interpretables y que tengan sentido físico.

El objetivo principal de esta tesis es diseñar diferentes maneras de codificar el conocimiento físico para proporcionar métodos de aprendizaje automático adaptados a problemas específicos en teledetección. Introducimos nuevos métodos que pueden fusionar de manera óptima fuentes de datos heterogéneas, explotar las regularidades en los datos, incorporar y respetar ecuaciones diferenciales, obtener modelos precisos que emulan y son coherentes con modelos físicos, y modelos que aprenden parametrizaciones del sistema combinando modelos y simulaciones.

1. Introduction

This thesis is concerned with the incorporation of domain knowledge in machine learning models in order to improve biophysical parameter retrieval using remote sensing data - both from observations and simulations from radiative transfer models. This chapter provides the reader with a brief introduction to remote sensing for biophysical parameter retrieval and the general approaches applied in the field. Subsequently, the machine learning algorithms used for parameter retrieval are discussed, as well as the ways in which expert knowledge can improve such algorithms. Finally the main research objectives, and the methods used to address them, are described.

1.1 Introduction to remote sensing

In its most literal sense, remote sensing (RS) refers to all information acquisition about an object from afar, and thus includes the use of handheld sensors as well as those based on aircrafts and satellites. It is in particular the satellite-based remote sensing systems that have allowed us to monitor ocean, land and atmosphere at a global scale and derive key insights about the climate.

Dependent on the *radiation source* involved in the data acquisition, remote sensing imaging instruments are partitioned into two categories: *passive* sensors, which rely on solar radiation as the illumination source (Ustin, 2004; Liang, 2004), and *active* sensors, where the energy is emitted by an antenna towards the Earth's surface and the energy scattered back to the satellite is measured (Mott, 2007; Wang, 2008). Some examples of passive sensors are infrared, charge-coupled devices, radiometers, passive microwave, and multi and hyperspectral sensors (Shaw & Manolakis, 2002). On the other hand, Radar systems, such as Real Aperture Radar (RAR) and Synthetic Aperture Radar (SAR), are examples of systems for active remote sensing. The data used in this thesis are derived from optical multispectral, infrared sounder and passive microwave sensors. Nevertheless, we will mainly focus on optical satellite sensors and simulations thereof.

Optical sensors record data in different bands at different wavelengths of incoming photons. If a sensor has 3-20 relatively wide bands it is said to be *multispectral*, while

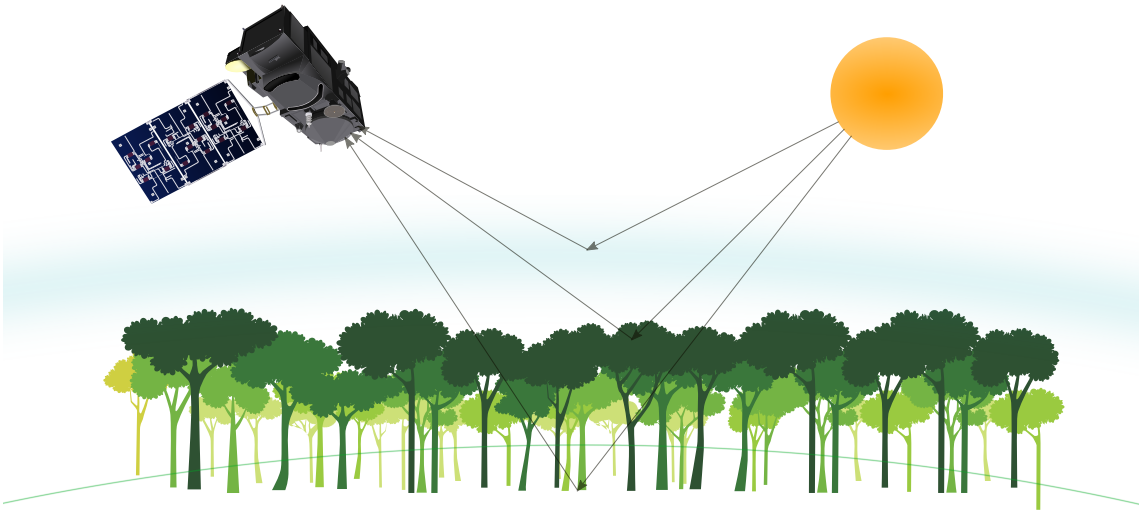


Figure 1.1: Remote sensing of the Earth's surface from a satellite sensor. The light of the sun interacts with the different layers of the atmosphere and the vegetation of the planet. Depending on their molecular structure, they absorb and scatter light at certain wavelengths which leaves a "spectral fringerprint" in the observed reflectance that can be used to make inference about the physical state of the earth.

sensors with hundreds of narrower bands is said to be *hyperspectral*. The electromagnetic radiation that leaves the Earth, measured at a high spectral resolution, is a rich source of information about the various processes that take place on the planet, and has a long list of applications. Among these applications we find 1) Waterbody monitoring: Sea surface salinity estimation (Lagerloef et al., 1995), water quality assessment (Ruescas et al., 2018) and sea ice monitoring (Spreen et al., 2008), 2) Land monitoring: Crop yield prediction (Mateo-Sanchis et al., 2019), drought detection (Kogan, 1995) and estimation of vegetation-based carbon uptake (Alton et al., 2007), and 3) Monitoring the constituents of the atmosphere: CO_2 emission and absorption (Tramontana et al., 2016), cloud detection (Mateo-García et al., 2017) and ozone monitoring (Kondratyev & Varotsos, 2002).

Earth observation through remote sensing data is a multi-disciplinary field incorporating, among others, physics, chemistry, geology, biology and computer science. Studying how electromagnetic radiation is reflected, absorbed, and emitted by different gasses, liquids and solids allows us to make inference about the material composition of the scene that we have acquired spectral information about. One of the tools that have emerged from such studies and served to advance them further is the *Radiative Transfer Model* (RTM), which simulates the transfer of electromagnetic radiation through our planetary atmosphere.

1.1.1 Radiative transfer models (RTMs)

RTMs describe the complex interactions of scattering and absorption of radiation with the constituents of the atmosphere, water, vegetation and soils. RTMs are useful because they allow us to translate (map) a set of parameter values describing the state of soil, leaf, canopy and atmosphere to at-sensor reflectance or radiance. Such simulations allow for modelling, understanding, and predicting parameters related to the state of the land cover,

water bodies and atmosphere.

Throughout this thesis we have made use of the specific RTM called PROSAIL. It is the combination of leaf optical properties model PROSPECT and canopy bi-directional reflectance model SAIL (Baret et al., 1992). PROSAIL is the most widely used RTM in the last twenty years in remote sensing studies (Jacquemoud et al., 2009). It mimics canopy reflectance using the turbid medium assumption (i.e., assuming the canopy as a turbid medium for which leaves are randomly distributed), which is particularly well suited for homogeneous canopies. PROSAIL simulates reflectance at wavelengths between 400 and 2500 nm with a 1 nm spectral resolution as a function of several parameters that account for the biochemistry and structure of the canopy, its leaves, the background soil reflectance and the sun-sensor geometry. In particular:

1. *Leaf optical properties*, given by the mesophyll structural parameter (N) and leaf chlorophyll (Chl), dry matter (Cm), water (Cw), carotenoid (Car) and brown pigment (Cbr) contents.
2. *Canopy level characteristics*, determined by leaf area index (LAI), the average leaf angle inclination (ALA) and the hot-spot parameter (Hotspot). System geometry is described by the solar zenith angle (θ_s), view zenith angle (θ_v), and the relative azimuth angle between both angles ($\Delta\Theta$).

Using the spectral response function of a given satellite-sensor, the PROSAIL output can be projected onto a lower-dimensional space in order to simulate how the satellite-sensor would record the reflectance.

RTMs encode the *forward direction* of the remote sensing problem, i.e. predicting the observed reflectance given the underlying state of the physical system. A key task of remote sensing is to solve the *inversion problems* of trying to predict the physical state, given the observed reflectance. For a visual summary, see Fig. 1.2.

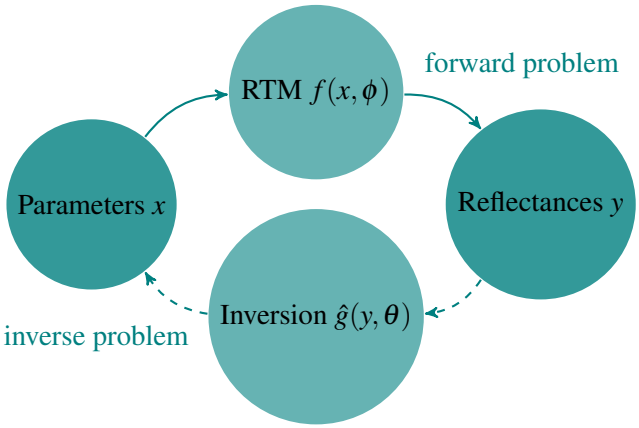


Figure 1.2: The forward problem in Earth observation involves taking the physical state of the system as input, defined by representative biophysical parameters (e.g. vegetation canopy or leaf characteristics), then propagating the solar radiation through the atmosphere medium and producing a simulated at-sensor reflectance. The inverse problem involves performing inference over the forward model f which is an RTM in this case. In other words predicting the underlying physical state parameters $\mathbf{x}$, given the observed reflectance $\mathbf{y}$. Both the forward RTM f and the inverse retrieval model $\hat{g}$ are complex nonlinear functions defined by their parameters ϕ and $\boldsymbol{\theta}$.

1.1.2 Biophysical parameter retrieval

As previously mentioned, remotely sensed data are applied in many subfields of Earth observation. One of the most central problems is that of biogeophysical parameter estimation which is the type of problem we are mainly concerned with in the present work. This refers to leaf- and canopy-level parameters of vegetation such as Chlorophyll content and leaf area index (LAI) respectively, meteorological variables such as air temperature and humidity and soil moisture, or ocean colour variables such as coloured dissolved organic matter (CDOM) and inorganic suspended matter (ISM).

The types of algorithms first used for parameter retrieval were algorithms based on expert knowledge of the physics behind the processes taking place in remote sensing. An example of this is the vegetation indices based on the fact that healthy green leaves with high levels of chlorophyll absorb visible light and reflects near infrared light strongly[1]. RTMs encode the equations we believe govern the physical system, and parameter retrieval has therefore also been referred to as inverting RTMs. A commonly used method for model inversion is to use an RTM to generate a representative database of input-output pairs, a so-called look-up table (LUT), and match a recorded spectrum with the closest one in the LUT according to some metric.

In the last two decades, another paradigm which has grown very popular is the use of data-intensive methods which are largely model agnostic. These *machine learning* methods rely on having adequate amounts of sufficiently high-quality data in order to learn a mapping from biophysical variable to observed spectrum. While there are vast amounts of satellite data available, labeled data is scarce. This has popularized the use of RTM to generate the datasets necessary to train the statistical models.

1.2 Machine learning for biophysical parameter retrieval

The earliest approaches to mapping observed reflectances to biophysical parameters were based on vegetation indices. A vegetation index is a transformation of bands of a satellite measurement based on knowledge about the interaction between photons and vegetation. These indices were used in conjunction with linear regression to monitor vegetation as early as 50 years ago (Jordan, 1969). Some of the most effective vegetation indices are based on simple normalized differences between bands (Rouse et al., 1974) and are still widely used in many remote sensing applications. The next step was to take as much spectral information into account as possible. Approaches based on calculating both derivatives (Peñuelas et al., 1994) and integrals (Broge & Leblanc, 2001) of specific spectral regions were proposed. Other approaches simply used all bands in an observed reflectance and applied step-wise multiple linear regression in order to remove bands with weak correlation from the regression (Faurtyot & Baret, 1997). The number of dimensions can be prohibitive for learning when using all bands, especially in the case of hyperspectral measurements, which led to the widespread use of principal component analysis (PCA) regression (Wold et al., 1987).

It has been shown that the relation between biophysical parameters and satellite spectra is non-linear (Camps-Valls et al., 2011). Therefore, unless you have perfectly handcrafted features, e.g. vegetation indices, you cannot find a linear combination of input variables that predict your output variable well. It is impossible to engineer features which will

[1] The most popular example of this is the normalized difference vegetation index (NDVI), see (Rouse et al., 1974).

linearize every problem for every sensor. Therefore, in the last two decades, the main focus in the field of parameter retrieval has been to use machine learning to perform a *non-linear mapping from the observed reflectance to the parameter of interest.* Below we provide a non-exhaustive summary of some of the most popular approaches.

Neural network methods are commonly used ML algorithms in the remote sensing literature, especially in recent years (Zhu et al., 2017). They scale well to large datasets ($\mathcal{O}(n)$ where n is the number of training data) and are able to model complex functions. These algorithms have enjoyed widespread popularity in the field of machine learning since the early 90s, leading to breakthroughs which have subsequently been used in the field of remote sensing. The most obvious example is the convolutional neural network whose unique characteristics makes it highly suitable for processing multiband remote sensing image data (Malmgren-Hansen et al., 2019). These algorithms, however, need a substantial amount of training data in order to avoid overfitting which is not always available in remote sensing problems.

Another well-studied algorithm in the machine learning literature is that of random forest regression (RFR), popular for its robustness and its ability to handle different data modalities and missing values. For these reasons, and the fact that they scale linearly with the amount of training data as well, RFR has been used in many parameter retrieval scenarios (Adam et al., 2014; Li et al., 2014; Tramontana et al., 2016; Moreno-Martinez et al., 2018). They furthermore have a way to rank data features which is very useful when trying to select the most important features for prediction.

Different from the two above-mentioned methods, a Gaussian Process (GP) is a probabilistic model which makes inference by maximizing the marginal likelihood, and which results in a predictive distribution (Rasmussen, 2003). This has made GPs very attractive in remote sensing applications as they provide not only predictions on biophysical parameters but also uncertainty estimates, allowing users to assess the quality of a prediction. GPs are good at handling small datasets with high dimensionality without overfitting, which is a typical scenario in remote sensing. Recent developments, however, have removed this barrier, allowing them to train on large datasets (Salimbeni & Deisenroth, 2017) as we will show later for temperature and moisture retrieval problems. Like RFR, there are ways of performing feature ranking with GPs which, together with the above reasons, have made them one of the most used ML algorithms in remote sensing (Camps-Valls et al., 2016). While Gaussian processes model the output values as stochastic, it is also possible to assume a random distribution over the input variables. Using the Bayes rule you can then derive a posterior distribution over your biophysical variables which usually has to be sampled from using Monte Carlo sampling methods, such as the popular Markov chain Monte Carlo (MCMC) algorithms (Martino & Elvira, 2017; Martino et al., 2018; Robert & Casella, 2013). The advantage of this type of modelling is the ability to model multi-modal posterior distributions. This corresponds to the case when different configurations of biophysical parameters result in the same observed reflectance spectrum which is common in remote sensing (Gómez-Dans et al., 2016).

1.3 Incorporating physics in machine learning

The last decade has seen a large increase in papers published in the cross-field of physics based modelling and machine learning (Willard et al., 2020). The various lines of research

implied include using ML to solve partial differential equations (Geneva & Zabaras, 2020), discovering governing equations (Raissi et al., 2017) and physics-guided data-generation (Yang et al., 2019). Particularly relevant for parameter retrieval in remote sensing, however, are the research topics of 1) improving machine learning regression by encoding physics knowledge, and 2) emulation of physics-based computer models. The methods presented in this thesis fall into these two categories, described below in more detail.

Improving machine learning regression by encoding physics knowledge

Machine learning methods are able to learn complex relationships from data, but have no concepts of physically meaningful output values, energy balances or boundary conditions. Encoding such knowledge in a regression method has been shown to improve predictions (Svendsen et al., 2017; Willard et al., 2020), especially outside domains of ample training data. Furthermore, depending on how the physics knowledge is encoded in the ML model, increased interpretability of learned parameters can be achieved.

One approach is to select an existing ML method and modify the cost function which is optimized in the learning process. An example of this is the physics-guided neural network (Karpatne et al., 2017), which apart from the standard error- and regularization-terms adds a term which penalizes deviations between NN predictions and predictions of a simple, yet robust and physically meaningful model. An example of this type of regression algorithm is presented in Sec. 3.1.

As opposed to designing a *cost function* that enforces coherence with principles of physics, other approaches build the ML *model* itself around the governing equations of a system. An example of this is the latent force model (LFM) (Alvarez et al., 2009), which assumes that the forcing in an ordinary differential equation (ODE) is governed by a Gaussian process, which in turn leads to GP solution to the ODE. Learning the hyperparameters of said GP corresponds to learning the parameters of the ODE as well as more physically consistent model predictions. An application of this method is presented in Sec. 3.2.

There are other frameworks that can be used to encode domain knowledge in ML regression, based on neural networks, that will not be considered in this thesis. Nevertheless, they constitute a very interesting research direction. One approach is to perform pre-training of a network on simulated data followed by a fine-tuning of parameters using real data (Jia et al., 2019). Another approach, presented in (Daw et al., 2020), is to build a constraint of monotonous increase in the modelled physical quantity into the recurrent NN architecture. It has also been shown that expert knowledge can be used to search for optimal NN architectures (Ba et al., 2019). For a review of such methods, see (Willard et al., 2020).

Emulation of physics-based computer models

Computer code simulations which act as convenient approximations to reality are ubiquitous in physics, brain, social, Earth and climate sciences. Such simulations allow us to model, understand, and predict parameters of interest. They do, however, often come with drawbacks such as a high computational cost and mathematical intractability of derivatives and integrals (Rivera et al., 2015; Gustau Camps-Valls, 2019). Training an ML algorithm on data generated by a simulator can ameliorate these shortcomings, resulting in an *emulator* which is faster to run, and, depending on the ML algorithm, differentiable. The increased computational speed allows otherwise prohibitively expensive sensitivity analy-

sis (Sobol, 1993) and the construction of arbitrarily large simulated datasets. The property of differentiability enables uncertainty propagation as well as the use of the emulator in cost functions of other ML methods. Section 4.1 presents work on the improvement on state-of-the-art emulation methods.

1.4 Research objectives

We propound that in order to truly advance the field of remote sensing data analysis it is necessary to combine physics and domain knowledge with advanced data-driven machine learning models. We therefore define the following goal:

> *"The overarching goal of this thesis is to explore new ways of improving data-driven algorithms by incorporating expert knowledge in order to solve forward and inverse modelling problems in remote sensing."*

This goal is pursued by exploring the interaction between ML and physics in Earth observation along two research directions. On the one hand, incorporating physical knowledge in machine learning regression algorithms to improve performance, consistency and interpretability. On the other hand, improving the quality of emulators of complex and often mathematically intractable physical models. These directions of research are both timely and challenging.

Why is the topic important?

The machine learning literature consists mainly of model-agnostic algorithms which are built to be as flexible as possible. This flexibility can lead to predictions which are not consistent with expert knowledge and may not respect the fundamental laws of physics, such as energy and mass conservation, or can even lead to meaningless predictions, e.g. negative estimates of strictly positive variables. By incorporating physical knowledge in machine learning regression it is possible to tailor a method to the particular problem at hand, making it more effective/consistent and credible. Emulators of physical models are important because they allow faster and more thorough exploration of the physical system in question. They also facilitate the use of physical models when building machine learning cost functions.

How do we plan to address it?

We propose probabilistic modelling, e.g. Gaussian processes and approximate Bayesian inference, as the appropriate framework for tackling forward and inverse problems in remote sensing. Several developed approaches will be introduced.

We show how to improve performance and interpretability for inversion in different ways: 1) combining observational data and simulations synergistically in GP regression, 2) using GP kernels derived from specific ODEs enforcing physical consistency, 3) performing approximate bayesian inference over radiative transfer models allowing us to recover probability distributions over physical parameters, and 4) learning faster and more flexible inversion models with deep GPs. In order to build more effective emulators we apply GPs in an active learning scheme that makes use of geometry and diversity information learned from the data thus leading to more compact and efficient emulators of physical simulators.

We will showcase the performance of the explored methods in various challenging remote sensing problems, involving a wide variety of satellite sensors (both optical and infrared sounder), biophysical parameters (leaf area index, chlorophyll content, soil moisture, or dissolved organic matter), Earth spheres (land/vegetation, ocean and the atmosphere), model simulations (from leaf-canopy and atmospheric models) and observational data (both remotely-sensed and in-situ). Each chapter provides a critical assessment of the method presented and a discussion of its relevance for forward and inverse problems in remote sensing.

1.5 Outline

The remainder of the thesis is organized as follows:

Chapter 2 reviews Gaussian processes, specifically for regression. The deep Gaussian process regression model is then presented for biophysical parameter retrieval, showing improved performance compared to existing models.

Chapter 3 describes two ways of encoding physics knowledge in Gaussian process regression: Firstly, one that jointly models in-situ and simulated data, improving regression on the in-situ data especially in scenarios of extrapolation. Secondly, one that models the data as the solution to an ODE with a Gaussian process forcing, allowing for physical interpretation of the fitted model.

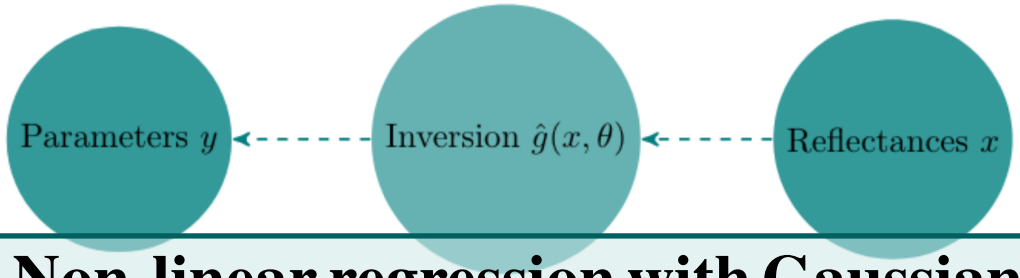
Chapter 4 presents an active learning framework for emulating an RTM. We see how an effective emulator can be built with a minimum number of evaluations of the RTM.

Chapter 5 reviews the variational autoencoder and the Monte Carlo expectation maximization algorithms. It then shows how to use each method to perform inference over an RTM model in order to derive probability distributions over biophysical parameters and compares strengths and weaknesses of the two approaches.

Chapter 6 summarizes the contributions of this thesis, discusses the main conclusions, and provides a list of related publications that resulted from the work performed during this PhD thesis.

Chapter 7 provides an overview of the thesis in Valencian.

We end the document with an Appendix section that contains the peer-reviewed scientific publications directly related to the work conducted in this thesis.



2. Non-linear regression with Gaussian processes

As explained in Sec. 1.2 the relation between biophysical parameters and satellite spectra is non-linear. Non-linear regression is therefore a key part of parameter retrieval. The methods in this thesis are largely based on Gaussian process regression. This is due to the fact that it is a flexible probabilistic regression model which offers intuitive ways to encode prior knowledge. This section provides an introduction to the GP regression method which is already widely used for parameter retrieval in RS settings (Camps-Valls et al., 2016). We furthermore present the use of deep Gaussian process regression, first developed by (Damianou & Lawrence, 2013), for improved parameter retrieval. Consistent with the machine learning literature we use letters x and y to refer to input and output respectively. Thus, when dealing with inverse problems, we will use x for reflectances and y for biophysical parameters.

This chapter is partly based on the publication:

1. **Svendsen, D.H.**, Morales-Álvarez, P., Ruescas, A.B., Molina, R. and Camps-Valls, G., 2020. Deep Gaussian processes for biogeophysical parameter retrieval and model inversion. *ISPRS Journal of Photogrammetry and Remote Sensing, 166*, pp.68-81.

2.1 Gaussian process regression

In machine learning, regression is a supervised learning problem where we try to find a mapping from input space $\mathcal{X}$ to output space $\mathcal{Y}$. We shall, for now, concern ourselves with single dimensional output values so that $\mathcal{X} = \mathbb{R}^D$, $D \in \mathbb{Z}^+$ and $\mathcal{Y} = \mathbb{R}$. We therefore consider a set of input-output datapairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and the corresponding input data matrix $\mathbf{X} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top, \ldots, \mathbf{x}_n^\top]$ and output data vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top$. This dataset, coupled with some inference scheme, can then be used to fit the parameters of a non-linear regression algorithm, in this case Gaussian process regression.

A Gaussian process is a *probability distribution over functions*. As opposed to a Gaussian distribution over a random variable, being defined by its constant mean vector and covariance matrix, a GP is determined by its *mean* and *covariance functions*:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \end{aligned}$$

and we write a function f determined by a Gaussian process as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{2.1}$$

Now, while the mean and covariance functions of a GP are actually *defined* everywhere in input space $\mathcal{X} = \mathbb{R}^d$, the useful properties of a GP really emerge when considering a *finite* number of evaluations of f:

"A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution" (Rasmussen, 2003)

This means that the vector of evaluations of f in our n training-inputs $\mathbf{f} = f(\mathbf{X}) = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)]^\top$ follows a multivariate normal distribution

$$f(\mathbf{X}) \sim \mathcal{N}(\mathbf{m}, \mathbf{K}) \tag{2.2}$$

with mean vector $\mathbf{m} = m(\mathbf{X}) = [m(\mathbf{x}_1), m(\mathbf{x}_2), \ldots, m(\mathbf{x}_n)]^\top$ and covariance matrix $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$ with elements $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The covariance function $k(\mathbf{x}, \mathbf{x}')$, which will be described in more detail in Section 2.1.1, is a function that encodes the similarity between values of f. If we know, for example, that the function is periodic, this can be reflected in the covariance, or kernel, function. The choice of the form of $k(\mathbf{x}, \mathbf{x}')$ is the most important choice when modelling data with GPs and training the model amounts to fitting the hyperparameters of the covariance function $\boldsymbol{\theta}$ and the noise variance σ^2. The mean function $m(\mathbf{x})$ can be used to encode prior information about the behaviour of f but is often assumed to be zero if no such information is available. In the following we will assume $\mathbf{m} = \mathbf{0}$.
Since the prior over latent function values is Gaussian (see Eq. (2.2)) we know that if we assume a Gaussian likelihood model

$$\mathbf{y} = f(\mathbf{X}) + \mathbf{e}\,, \quad \mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \tag{2.3}$$

then we can easily compute the marginal likelihood (see (Bishop, 2006) Eq. (2.115)) which

becomes

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}). \tag{2.4}$$

We can maximize the log of the marginal likelihood with respect to the hyperparameters $\boldsymbol{\theta}$ and σ using gradient descent, which is the standard way of training GPs.

Now, what we are really interested in is regression, that is, in predicting an output value y_* given an observed input $\mathbf{x}_*$. The GP framework handles this by considering the joint distribution over the observed outputs $\mathbf{y}$ and the unobserved output y_*, given by

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k}_*^T \\ \mathbf{k}_* & k(\mathbf{x}_*, \mathbf{x}_*) + \sigma^2 \end{bmatrix}\right),$$

where $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_n)]^\top$ is an $n \times 1$ vector. Using standard manipulation of joint normally distributed variables (see (Bishop, 2006) Eq. (2.96-97)), we can obtain the distribution over y_* conditioned on the training data y, i.e.,

$$p(y_*|\mathbf{y}) = \mathcal{N}(\mu_{\text{GP}}, \sigma^2_{\text{GP}}).$$

This is a Gaussian distribution with predictive mean and variance given by

$$\mu_{\text{GP}}(\mathbf{x}_*) = \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \tag{2.5a}$$
$$\sigma^2_{\text{GP}}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) + \sigma^2 - \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_*. \tag{2.5b}$$

The fact that the GP not only provides predictions μ_{GP} for a given input but also has a natural way of assessing the uncertainty of a prediction through σ^2_{GP} has made it a very popular method in remote sensing (Camps-Valls et al., 2016).

2.1.1 Covariance functions

The covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ maps two points in the input space onto the real line. It encodes knowledge about the function we are trying to model in that it describes how similar two outputs $\mathbf{y}$ and $\mathbf{y}'$ should be, given their corresponding inputs $\mathbf{x}$ and $\mathbf{x}'$. A valid covariance function, or kernel function as it also called, always results in a positive definite covariance matrix with elements $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, constructed from any set $\{\mathbf{x}_i\}_{i=1}^n \; \forall \, \mathbf{x}_i \in \mathcal{X}$. Below are some examples of commonly used covariance functions.

Exponentiated Quadratic: This kernel function has the form

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{\lambda}\right),$$

where λ is the *lengthscale* of the function. The closer two points are in input-space, the stronger the covariance of their corresponding outputs. The EQ kernel results in an infinitely differentiable predictive function.

Matérn: This is a family of kernels (for the general formula see, e.g., (Schölkopf et al., 2002)) that are used to model less smooth functions that are not infinitely differentiable.

One of the most used kernels in the family is the Matérn $3/2$ kernel which has the form

$$k(\mathbf{x},\mathbf{x}') = \left(1+\frac{\sqrt{3}\|\mathbf{x}-\mathbf{x}'\|}{\lambda}\right)\exp\left(\frac{-\sqrt{3}\|\mathbf{x}-\mathbf{x}'\|}{\lambda}\right),$$

where λ is the *lengthscale*. This kernel function is once-differentiable and is useful for modelling non-smooth functions.

Periodic: The periodic kernel, as derived by D. MacKay (MacKay, 1998), is useful when we know that the function has a certain period. It takes the form

$$k(\mathbf{x},\mathbf{x}') = \exp\left(\frac{-\sin^2(2\pi\|\mathbf{x}-\mathbf{x}'\|/p)}{\lambda}\right)$$

where p is the period. There are variants which allow for the period and amplitude to change across the input space. Figure 2.1 visualizes the different kernel functions and shows examples of function values sampled from the GP priors that the above kernel functions give rise to.

The above-mentioned kernels are all examples of so-called *stationary kernels*, namely kernels that are functions only of the distance between inputs $d = \|\mathbf{x}-\mathbf{x}'\|$. This translation invariance is not always a good assumption to make, in fact non-stationary kernels have been shown to work better for various remote sensing problems (Gewali et al., 2019). Nevertheless, simple kernel functions such as the EQ-kernel are usually quite expressive and very useful for developing new algorithms. When the number of training data is sufficiently high, the EQ-kernel can approximate any function arbitrarily well (Rasmussen, 2003).

2.2 Improving GP regression for parameter retrieval with deep GPs

Gaussian processes are widely used in remote sensing (Camps-Valls et al., 2016), among other reasons, due to their ability to quantify predictive uncertainty and model non-linear data without needing big datasets. Classic GP regression, however, has two important drawbacks: On the one hand, the training cost scales cubically $\mathcal{O}(n^3)$ with the number of training samples n, which makes the method prohibitively expensive for larger dataset[1]. On the other hand, when it comes to inverting an RTM with a complex and hierarchical structure, a single layer GP with a fixed kernel form is not always able to model the data. Deep Gaussian processes, first proposed in (Damianou & Lawrence, 2013), have proven able to model more complex data than their shallow counterparts while possible to train at a cost that scales linearly with the number of training samples (Salimbeni & Deisenroth, 2017). The following explains the doubly stochastic variational inference scheme for DGPs and presents a comparison of DGPs with single layers GPs for parameter retrieval tasks.

[1] Datasets larger than 10 000 datapoints are usually considered too large for classic GP regression.

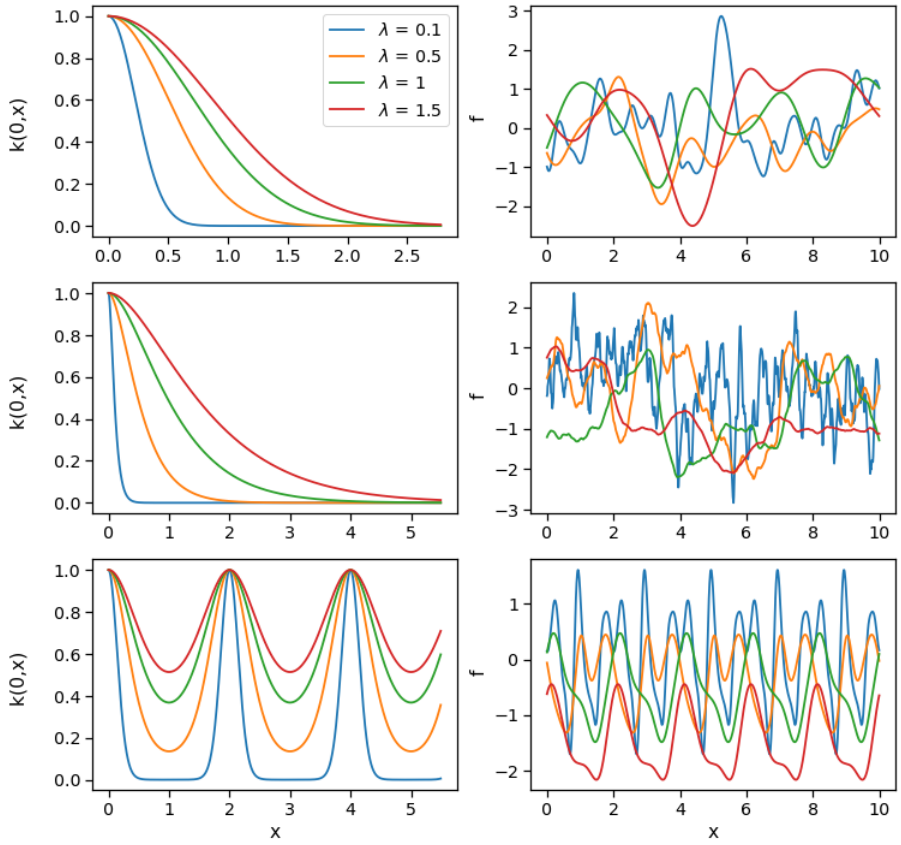


Figure 2.1: The left column visualizes different kernel functions as a function of distance for different values of lengthscale λ. From top to bottom the kernels are the Exponentiated Quadratic, the Matérn 3/2 and the Periodic kernel. The right column shows samples from the GP priors that the kernels on the left give rise to. The colors of the samples correspond to the those of the kernel function on the left.

2.2.1 Sparse Gaussian processes

Before the advent of DGPs, a wide range of approximations have been presented in the GP literature in order to overcome the above-mentioned drawback of a high computational cost (Snelson & Ghahramani, 2006; Hensman et al., 2013; Bauer et al., 2016; Morales-Alvarez et al., 2017). Most approximations rely somehow on a set of *inducing points* which are a reduced set of $m \ll n$ of latent variables. These inducing points $\mathbf{u} = (u_1, \ldots, u_m)$ are GP realizations at the *inducing locations* $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\} \subset \mathbf{R}^d$, in the same way that the GP realizations in $\mathbf{f}$ are at the inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.

Bauer et al. make a succinct partitioning of all these methods based on whether the approximation takes place in the model definition or in the inference procedure (Bauer et al., 2016). The *Fully Independent Training Conditional* (FITC) (Snelson & Ghahramani, 2006) is the most popular example of the former approach. It assumes an approximate GP model by introducing the inducing points and then marginalizes them out using exact inference. On the other hand, the *Scalable Variational Gaussian Process* (SVGP) is the most popular example of the approaches which maintain the exact GP model. The SVGP introduces the inducing points as parameters of a variational distribution, making an approximation in the inference scheme. This variational inference scheme has proven very

useful for DGPs since the posterior is analytically intractable as will be explained below. In fact, the inference scheme used here for the DGP reduces to that of the SVGP when the depth of the DGP is $L = 1$. The experiments of Sec. 2.2.4 compare the performance of the DGP with the FITC and the SVGP, which both have linear training costs, as well as the standard GP for parameter retrieval. For more details on the FITC method, see (Svendsen et al., 2020b) or (Snelson & Ghahramani, 2006).

Table 2.1: Summary of the main differences between the four GP-based models used in (Svendsen et al., 2020b). VI = Variational Inference, D = dimension of each layer, n_b = minibatch size.

	GP	FITC	DGP ($L = 1$)	DGP ($L > 1$)
References	2003	2006	2013; 2017	2017
Model	Exact	Approx.	Exact	Exact
Inference	Exact	Exact	Approx. (VI)	Approx. (VI)
Depth	Shallow	Shallow	Shallow	Deep
Training cost	$\mathcal{O}(n^3)$	$\mathcal{O}(nm^2)$	$\mathcal{O}(n_b m^2)$	$\mathcal{O}\left(n_b m^2 \sum_{l=1}^{L} D^l\right)$

2.2.2 Deep Gaussian process model

When applying standard (single-layer) GP regression, the GP output is directly used to model the response variable $\mathbf{y}$. This output could, however, be used to define the input position of another GP. Performing this stacking a total of L times yields a *Deep Gaussian Process* of L layers. Intuitively, this implies a richer generative model than that of the standard GP, able to capture more complex patterns in the data. Figure 2.2 shows samples drawn from a single-layer GP as well as from 2- and 3-layer DGPs. The bottom subfigure shows that the DGP is able to model functions exhibiting different lengthscales of variability in the same region as well as varying lengthscales across the input space.

For a standard GP, the Gaussian prior $p(\mathbf{f})$ is conjugate to the Gaussian likelihood model $p(\mathbf{y}|\mathbf{f})$. This means that one can integrate out $\mathbf{f}$ and compute the marginal likelihood $p(\mathbf{y})$ and the posterior $p(\mathbf{f}|\mathbf{y})$ in closed form (parameters are omitted for simplicity). For the DGP model, where latent values to be integrated out appear as inputs in the subsequent layer (i.e. they appear inside a complex covariance matrix), exact inference is intractable. For this reason, we introduce m inducing points $\mathbf{u}^l$ at inducing locations $\mathbf{z}^{l-1}$ at each layer l. The assumed form of the variational posterior over $\{\mathbf{f}^l\}_{l=1}^L$ and the inducing output $\{\mathbf{u}^l\}_{l=1}^L$ is what leads to tractability and a training cost that scales linearly with the number of training data. The rightmost plot in Fig. 2.3 shows a graphical representation of the described model. For notational simplicity, in the following, the dimensions of the hidden layers will be fixed to one (this can be generalized straightforwardly, see both (Damianou & Lawrence, 2013; Salimbeni & Deisenroth, 2017)). For observed $\{\mathbf{X}, \mathbf{y}\}$ the joint DGP regression model is

$$\mathrm{p}(\mathbf{y}, \{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L) = \mathrm{p}(\mathbf{y}|\mathbf{f}^L) \prod_{l=1}^{L} \mathrm{p}(\mathbf{f}^l|\mathbf{u}^l; \mathbf{f}^{l-1}, \mathbf{z}^{l-1}) \mathrm{p}(\mathbf{u}^l; \mathbf{z}^{l-1}). \tag{2.6}$$

Here, $\mathbf{f}^0 = \mathbf{X}$, and each factor in the product is the joint distribution over $(\mathbf{f}^l, \mathbf{u}^l)$ of a GP in the inputs $(\mathbf{f}^{l-1}, \mathbf{z}^{l-1})$, but rewritten with the conditional probability given $\mathbf{u}^l$. Notice that a semicolon is used to specify the inputs of the GP. We will now see how to use the

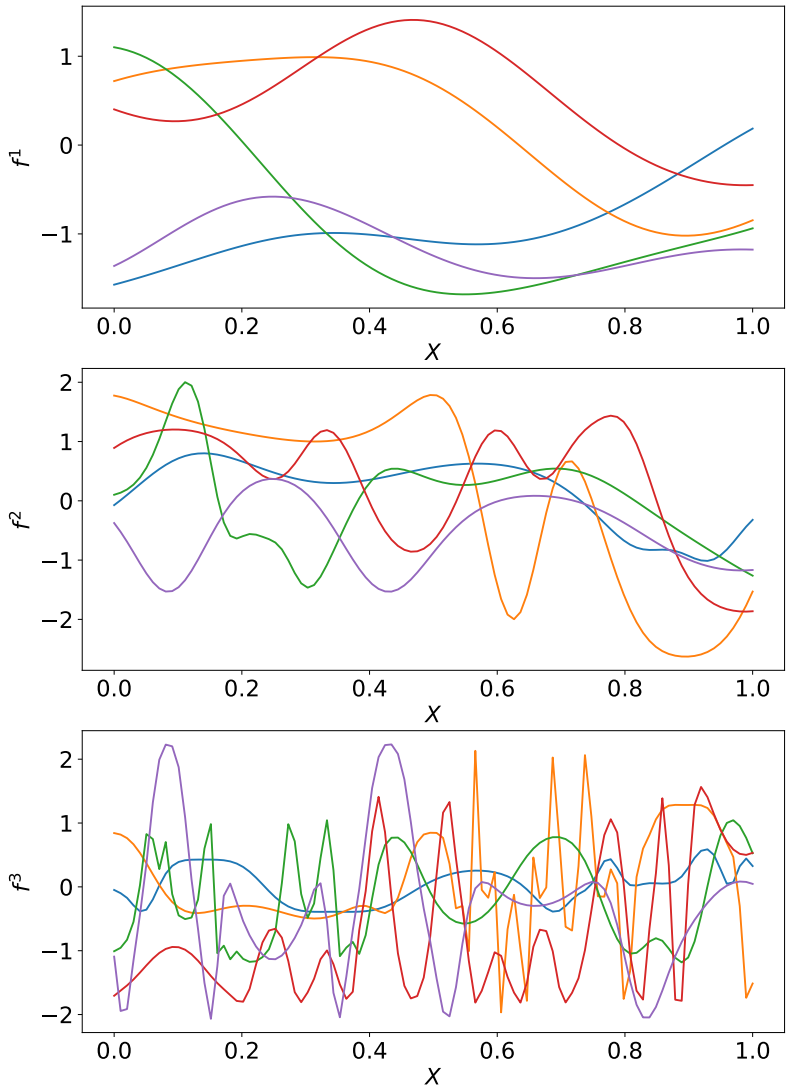


Figure 2.2: Five random samples from a 1-dimensional DGP with one (standard GP), two and three layers and one hidden unit per layer. Each function sample uses the function of the same color in the previous plot as input, except the function samples of the top plot ($L = 1$) which use the actual values of x as input. Every layer is endowed with a standard EQ kernel. This produces very smooth functions in the first layer (i.e. a shallow GP, top plot). However, the concatenation of such simple GPs produces increasingly complex functions (middle and bottom plots). In particular, notice that the 3-layer DGP captures sophisticated patterns that combine flat regions with high-variability ones, which cannot be described by stationary kernels.

inducing variables in a variational inference scheme for the DGP model.

2.2.3 Doubly stochastic variational inference

Variational inference (VI) is a widely applied approach in probabilistic modelling when computation of the posterior is intractable. It works by introducing a parametric family of candidate posterior distributions within which one seeks the optimal distribution. Since finding the analytical posterior involves a complicated process of integrating latent values out, it is said that VI transforms the integration problem to an optimization one. The

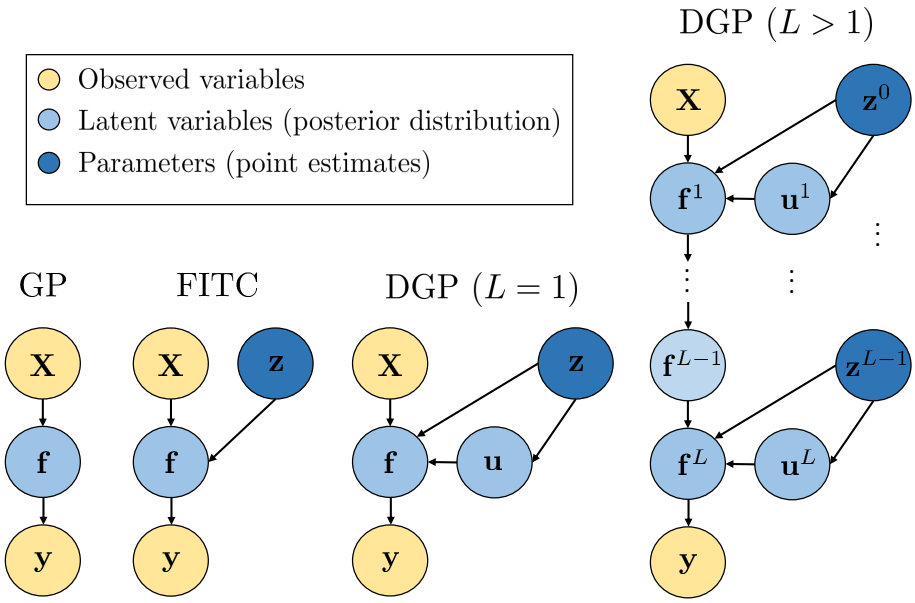


Figure 2.3: Graphical representation of the four GP-based models used in this work. The color indicates whether a variable is observed or must be estimated. In the latter case, the intensity of the color represents the type of estimation: either through a posterior distribution (light), or a point value (dark).

variational posterior family suggested in (Salimbeni & Deisenroth, 2017) is

$$\mathrm{q}(\{\mathbf{f}^l,\mathbf{u}^l\}|\{\mathbf{z}^l,\mathbf{m}^l,\mathbf{S}^l\}) = \prod_{l=1}^{L} \mathrm{p}(\mathbf{f}^l|\mathbf{u}^l;\mathbf{f}^{l-1},\mathbf{z}^{l-1})\mathrm{q}(\mathbf{u}^l) \tag{2.7}$$

where the set notation ranges from $l = 1$ to $l = L$. Notice here that the first term is conditional distribution of latent function values $\mathbf{f}^l$ given $\mathbf{u}^l$ of the *original DGP prior*. The second term is a Gaussian over the inducing outputs with mean $\mathbf{m}^l$ and full covariance $\mathbf{S}^l$ (which are variational parameters of the parametric family, to be estimated). In VI, the variational posterior is introduced in the expression of a lower bound to the evidence (or marginal likelihood) and is then optimized in order to find the best candidate of the variational family (Blei et al., 2017). The *evidence lower bound* (ELBO) in this case takes the form

$$\log \mathrm{p}(\mathbf{y}) = \log \int \frac{\mathrm{q}(\{\mathbf{f}^l,\mathbf{u}^l\})}{\mathrm{q}(\{\mathbf{f}^l,\mathbf{u}^l\})} \mathrm{p}(\mathbf{y},\{\mathbf{f}^l,\mathbf{u}^l\})\mathrm{d}\{\mathbf{f}^l,\mathbf{u}^l\} \geq$$
$$\sum_{i=1}^{n} \mathbb{E}_{\mathrm{q}(f_i^L)}[\log \mathrm{p}(y_i|f_i^L)] - \sum_{l=1}^{L} \mathrm{KL}(\mathrm{q}(\mathbf{u}^l)\|\mathrm{p}(\mathbf{u}^l;\mathbf{z}^{l-1})). \tag{2.8}$$

Due to the proposed variational posterior (2.7) the conditional of $\mathbf{f}^l$ given $\mathbf{u}^l$ cancels out and after some re-arranging and using the fact that probability distributions integrate to 1, the above result is obtained. The Kullback-Leibler divergence between two Gaussians is easy to compute analytically, but this is not the case for the expected value with respect to the marginals of the posterior at the last layer $\mathrm{q}(f_i^L)$. The idea here, then, is to sample from

the marginals and approximate the expected value. This is possible since the variational posterior is a multiplication of Gaussians making it easy to marginalize out the $\mathbf{u}^l$ values and obtain

$$\mathrm{q}(\{\mathbf{f}^l\}) = \prod_{l=1}^{L} \mathrm{q}(\mathbf{f}^l|\mathbf{m}^l,\mathbf{S}^l;\mathbf{f}^{l-1},\mathbf{z}^{l-1}) = \prod_{l=1}^{L} \mathcal{N}(\mathbf{f}^l|\tilde{\boldsymbol{\mu}}^l,\tilde{\boldsymbol{\Sigma}}^l), \tag{2.9}$$

where the vector $\tilde{\boldsymbol{\mu}}^l$ is a function $[\tilde{\boldsymbol{\mu}}^l]_i = \mu_{\mathbf{m}^l,\mathbf{z}^{l-1}}(f_i^{l-1})$ and the $n \times n$ matrix $\tilde{\boldsymbol{\Sigma}}^l$ by $[\tilde{\boldsymbol{\Sigma}}^l]_{ij} = \Sigma_{\mathbf{S}^l,\mathbf{z}^{l-1}}(f_i^{l-1}, f_j^{l-1})$. The explicit expression for the functions $\mu_{\mathbf{m}^l,\mathbf{z}^{l-1}}$ and $\Sigma_{\mathbf{S}^l,\mathbf{z}^{l-1}}$ can be found in (Salimbeni & Deisenroth, 2017, Eqs. (7-8)). The key point here is to observe that, although the distribution in Eq. (2.9) is fully coupled between layers (and thus the posterior in the last layer is analytically intractable), the i-th marginal at each layer $\mathcal{N}(f_i^l|[\tilde{\boldsymbol{\mu}}^l]_i, [\tilde{\boldsymbol{\Sigma}}^l]_{ii})$ only depends on the corresponding i-th input of the previous layer. This allows one to recursively sample $\hat{f}_i^1 \to \hat{f}_i^2 \to \cdots \to \hat{f}_i^L$ from all the layers up to the last one by means of just univariate Gaussians. Specifically, $\varepsilon_i^l \sim \mathcal{N}(0,1)$ is first sampled and then for $l = 1,\ldots,L$:

$$\hat{f}_i^l = \mu_{\mathbf{m}^l,\mathbf{z}^{l-1}}(\hat{f}_i^{l-1}) + \varepsilon_i^l \cdot \sqrt{\Sigma_{\mathbf{S}^l,\mathbf{z}^{l-1}}(\hat{f}_i^{l-1}, \hat{f}_i^{l-1})}. \tag{2.10}$$

The fact that we sample from $\mathrm{q}(f_i^L)$ in order to obtain an estimation of the expected value in the ELBO is the first source of stochasticity. And seeing as the ELBO factorizes over the training points allows us to obtain scalability by sub-sampling data in mini-batches which is the second source of stochasticity. Together, these methods motivate the name *doubly stochastic variational inference*. The ELBO is maximized with respect to the variational parameters $\mathbf{m}^l, \mathbf{S}^l$, the inducing locations $\mathbf{z}^l$, and the kernel and likelihood hyperparameters $\boldsymbol{\theta}^l$ (which, to alleviate the notation, have not been included in the equations). Notice that the complexity to evaluate the ELBO and its gradients is $\mathcal{O}\left(n_b m^2 (D^1 + \cdots + D^L)\right)$, where n_b is the size of the mini-batch used, and D_l is the number of hidden units in each layer (which were set to one in this section).

In order to predict in a new function value at $\mathbf{x}_*$, Eq. (2.10) is used to sample S times[2] from the posterior up to the $(L-1)$-th layer using the test location as initial input. This yields a set $\{f_*^{L-1}(s)\}_{s=1}^S$ of S samples. Then, the density over f_*^L is given by the Gaussian mixture (recall that all the terms in Eq. (2.9) are Gaussians):

$$\mathbf{q}(f_*^L) = \frac{1}{S}\sum_{s=1}^{S} \mathbf{q}(f_*^L|\mathbf{m}^L,\mathbf{S}^L;f_*^{L-1}(s),\mathbf{z}^{L-1}). \tag{2.11}$$

2.2.4 Experimental results

We now turn to the evaluation the model performance on a challenging remote sensing problem. The accurate estimation of atmospheric temperature and water vapour is essential for climate and weather forecasting studies. The Infrared Atmospheric Sounding Inter-

[2] This S is related to the first source of stochasticity and, theoretically, the higher the better. In practice, results become stable after a few samples.

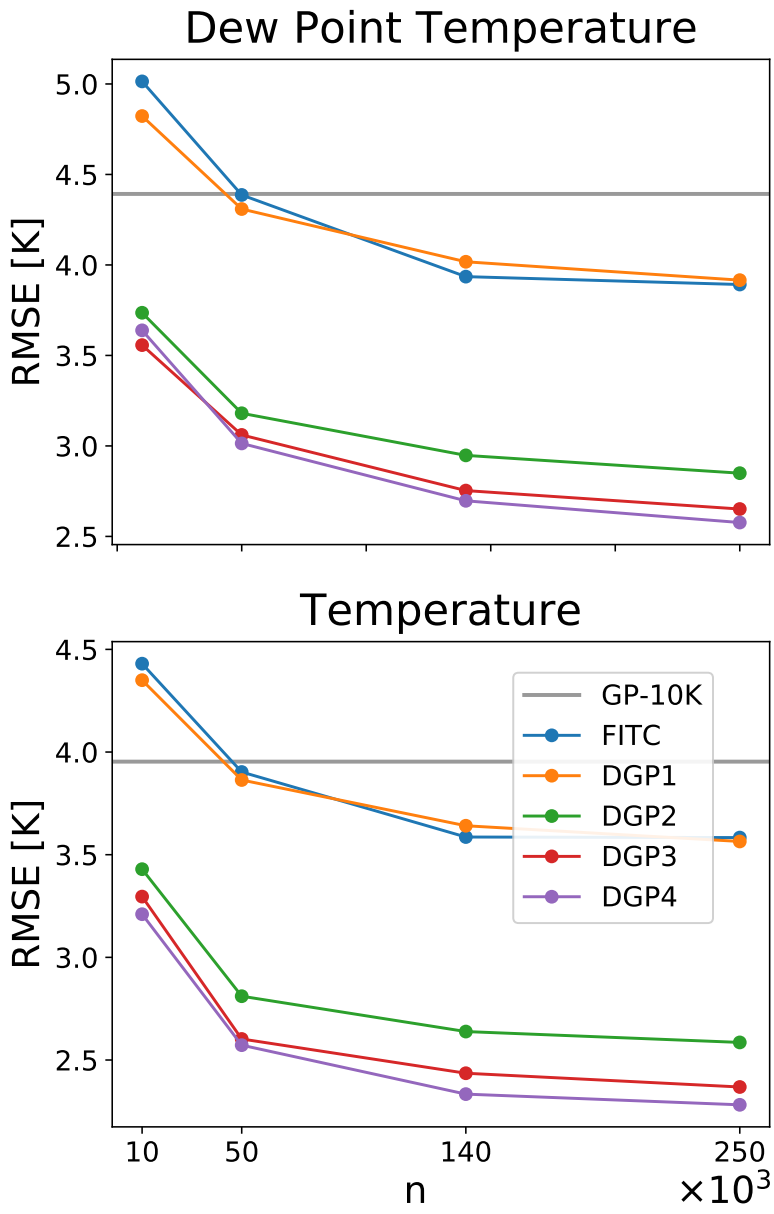


Figure 2.4: Performance of the compared methods as a function of the training set size for the surface dew point temperature (top) and temperature (bottom) variables. The RMSE of the Deep Gaussian Processes decreases with increasing depth. The deep models outperform the shallow ones which, only if given enough data, are able to outperform the GP-10K.

ferometer (IASI)[3] sensor implemented on the MetOp satellite series collects rich spectral information to derive temperature and moisture (Tournier et al., 2002). The temperature and moisture values are derived from the European Center for Medium-Range Weather Forecasts (ECMWF)[4] model. For details on data collection and preprocessing see (Svendsen et al., 2020b). We compare the following GP regression methods for retrieval of surface level temperature and dew point temperature (moisture):

[3] www.eumetsat.int/website/home/Satellites/CurrentSatellites/Metop/MetopDesign/IASI
[4] https://www.ecmwf.int/

DGP1-4: Doubly stochastic variational inference DGP with 1-4 layers and 300 inducing inputs per layer. The number of hidden units per layer is 5. Recall that DGP1 is equivalent to the sparse GP method SVGP, and the computational cost of DGP is $\mathcal{O}\left(n_b m^2 (D^1 + \cdots + D^L)\right)$.

FITC: Along with SVGP, it is the most popular sparse GP approximation. The EQ kernel is used, and the code is taken from GPflow[5]. The cost of training scales like $\mathcal{O}(nm^2)$, and the number inducing points is 300.

GP-10K: A standard GP using 10 000 training points is provided as a baseline. Recall that this is the limit of a standard GP in practice, since it scales like $\mathcal{O}(n^3)$. Again, the EQ kernel and the GPflow library are used.

The different methods are trained using datasets of sizes 10 000, 50 000, 140 000, and 250 000, and a testing set of 20 000. The accuracy in terms of root mean squared error (RMSE) shown in Fig. 2.4 is the average over five repetitions of the experiment.

What immediately stands out is the difference between the shallow (GP-10K, FITC, DGP1) and the deep models (DGP2-4). As intuitively expected, the performance of all models improves as they are allowed to leverage more training data. As the DGP1 and FITC models are only approximations of the standard GP, it is to be expected that they perform worse when training on the same amount of data, i.e. 10 000. Nevertheless, when allowed to leverage more data, their fit improves and outperforms the GP-10K. It is not clear which of the two approximations is superior, as it varies with the number of training data. This agrees well with the literature, where this has been shown to depend on the data at hand (Bauer et al., 2016). The fact that single-layer approximations can outperform a standard GP when given enough training data underlines the importance of a model which is able to handle large-scale data. We can see from the results that the DGP both handles large datasets but also allows for higher model complexity and thus a better fit of the data. From observing the performance of DGPs with different numbers of layers, we can see that DGPs take advantage of their hierarchical structure and achieve lower RMSE with increasing depth.

One of the most attractive properties of the GP regression is the natural estimate of predictive uncertainty given by the variance of the predictive distribution. The following provides a way to assess the quality of both the predictive mean and variance: According to the Gaussian predictive distribution, scaling the residuals of the predictive mean by the predictive standard deviation we obtain a variable $\zeta^* = \frac{\mu(\mathbf{x}^*) - y^*}{\sigma(\mathbf{x}^*)}$ which should follow a $\mathcal{N}(0,1)$. We can therefore plot the empirical densities of the scaled residual of each model and compare them to a $\mathcal{N}(0,1)$ distribution function. Figure 2.5 shows this plot using 10^6 test points to generate the empirical distributions for a DGP3 and FITC model both trained on 250 000 data-points and a GP trained on 10 000. Simply put, wide tails are the result of big residuals divided by small number (underestimation of predictive variance) while narrow distributions are the result of scaling by values that are too large (overestimation of predictive variance). We see that the scaled residuals of the DGP3 model best follow the $\mathcal{N}(0,1)$ distribution while the residuals of the FITC are skewed towards negative values and those of the standard GP are distributed very narrowly. These results indicate that the ability of the DGP to model more complex data leads to not only better predictive means,

[5] https://github.com/GPflow

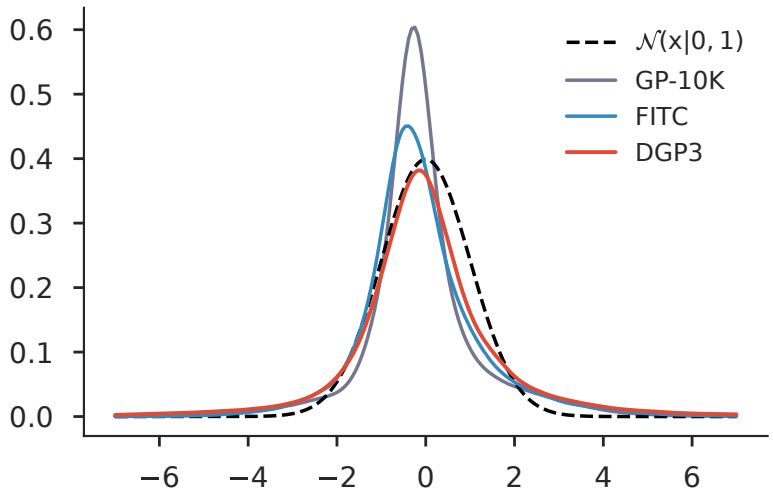


Figure 2.5: KDE of residuals normalized by predictive standard deviation, which according to the model should be standard normal distributed. The 3-layer DGP avoids the underestimation seen in the other models, and provides better estimates of predictive uncertainty.

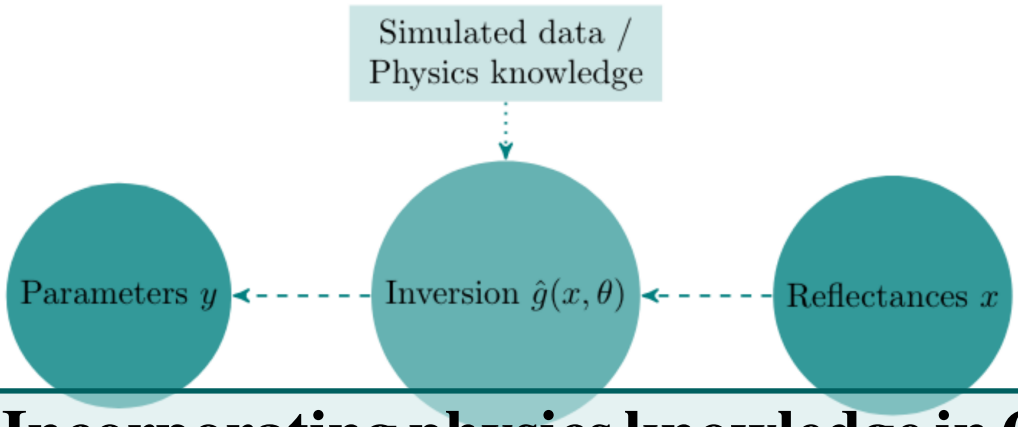
but also predictive uncertainties.

2.3 Concluding remarks

Gaussian process regression is a flexible, probabilistic model, and is one of the most popular algorithms for biophysical parameter retrieval (Camps-Valls et al., 2016). This chapter provided an introduction to GP regression as well as its extension: deep GP regression. We saw that DGPs are able to model more complex data-structures, and that they scale well to large numbers of data points. This ability to handle big and complex data while maintaining many of the properties of the single-layer GP such as predictive uncertainty estimation makes it a natural choice for parameter retrieval in remote sensing applications. In (Svendsen et al., 2020b) we also show how the DGP compares favorably to a deep neural network method for water-body monitoring applications.

The DGP model has demonstrated excellent performance in terms of accuracy and scalability, but future improvement is still needed. While the training cost of DGPs scale linearly with the number of training points, it is important to keep in mind that neural networks currently still maintain an edge when it comes to training speed. This is not surprising as the DGP is learning a predictive distribution instead of a single point estimate. Furthermore, when there is a clear spatial structure, convolutional neural networks are likely to be more effective (Malmgren-Hansen et al., 2019; Mateo-García et al., 2019), although there are some efforts in the direction of convolutional GPs (Van der Wilk et al., 2017).

In summary, GPs are widely used both for inversion and emulation of radiative transfer models (Camps-Valls et al., 2020), and DGPs are likely to improve performance on these important tasks in the field of remote sensing. This chapter has focused on improving non-linear regression with Gaussian processes in a remote sensing setting, which is a machine learning problem. The remaining chapters, however, will deal with the intersection between physical models and machine learning.



3. Incorporating physics knowledge in GP regression

There are different ways that one can improve regression algorithms through the incorporating physics knowledge (Willard et al., 2020). One approach is to incorporate simulated data from a physical model in the learning scheme of the ML algorithm. This can lead to improved performance but does not necessarily yield much physical insight. Another approach is to build an ML model based on knowledge of the governing equations of the system at hand. This tailors the regression method to the particular data and allows us to learn physical parameters in the training process of the ML model. The downside is that the assumptions of the model can be too strong, making it less flexible.

In this chapter, we shall see an example of each approach and how the incorporation of physics knowledge can lead to i) improved regression, and in particular extrapolation, and ii) insight into the underlying physical system at hand. In Sec. 3.1 we developed a GP framework that jointly models *in-situ* and simulated data and weighs training points based on the data source. In Sec. 3.2 we use the work on latent force models by (Alvarez et al., 2009) to tailor GP regression to the dynamics of remotely sensed soil moisture estimates. In doing so we learn a latent forcing that corresponds well with the (independently recorded) precipitation measured at the relevant site.

This chapter is partly based on the following publications:

1. **Svendsen, D.H.**, Martino, L., Campos-Taberner, M., García-Haro, F.J. and Camps-Valls, G., 2017. Joint Gaussian processes for biophysical parameter retrieval. *IEEE Transactions on Geoscience and Remote Sensing, 56(3)*, pp.1718-1727.
2. Camps-Valls, G., Martino, L., **Svendsen, D.H.**, Campos-Taberner, M., Muñoz-Marí, J., Laparra, V., Luengo, D. and García-Haro, F.J., 2018. Physics-aware Gaussian processes in remote sensing. *Applied Soft Computing, 68*, pp.69-82.

3.1 Joint Gaussian processes retrieval with in-situ and simulated data

Real *in-situ* measurements of biophysical parameters, which are matched with satellite data to yield the data-pairs we need to train machine learning models, are the result of expensive field campaigns. Due to the prohibitive cost and speed at which such data is gathered, there is a shortage of this type of data. On the other hand, simulations provide a fast and cheap way to obtain input-output pairs. We know, however, that the predictive distributions of real and simulated data tend to differ, because no physical model perfectly reflects the reality of the underlying physical system. It is important to model the real data because it reflects a real use-case, including possible regional and sensor-specific biases. The simulated data can be seen as an ideal, and thus less realistic, data-source which is easier to procure.

The model described here is designed to improve the regression on the real data by drawing predictive power from simulated data without learning undesirable aspects of the simulated dataset. It can be interpreted as a type of multitask Gaussian Process (Bonilla et al., 2008) which focuses on the primary task.

3.1.1 Model formulation

Let us now assume that the dataset $\mathcal{D}$ is formed by two disjoint sets: one set of r real reflectance-parameter pairs, $\mathcal{D}_r = \{(\mathbf{x}_i, y_i)\}_{i=1}^{r}$, and one set of s RTM-simulated pairs $\mathcal{D}_s = \{(\mathbf{x}_i, y_i)\}_{i=r+1}^{n}$, so that $n = r+s$ and $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_s$. In matrix form, we have $\mathbf{X}_r \in \mathbb{R}^{r\times D}$, $\mathbf{y}_r \in \mathbb{R}^{r\times 1}$, $\mathbf{X}_s \in \mathbb{R}^{s\times D}$ and $\mathbf{y}_s \in \mathbb{R}^{s\times 1}$. Finally, the $n\times 1$ vector $\mathbf{y}$ contains all the n outputs, sorted with the real data first, followed by the simulated data.

In order to improve prediction, one might consider pooling together the data into one dataset. This is a rather naive approach, however, as simulated and in-situ data tend not to follow the same distribution. As we shall see, this will often confuse a model rather than improve it. It is therefore important to incorporate a way to distinguish between the two datasets. Figure 3.1 shows a scenario where we have access to simulated data over the entire input space, which is often the case, and only access to real data in a limited range. Both datasets are generated from a damped exponential buried in white noise but the simulated data has a bias, reflecting the inability of computer-simulations to capture the behaviour of in-situ data. While the model that is trained on the smaller "in-situ" dataset performs poorly outside the region of data-availability the model following the naive approach of pooling the datasets indiscriminatively assigns too much importance to the simulated data-points. The model presented here prioritizes data from the main source, when available, and leverages data from the auxiliary dataset otherwise.

In order to distinguish the datasets we add a hyperparameter with respect to the standard Gaussian process, modelling noise of the simulated data relative to that of the real data

$$y_i = f(\mathbf{x}_i) + e_i, \qquad e_i \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix}\sigma^2 & \text{if} & i \le r \\ \sigma^2/\gamma & \text{if} & i > r\end{matrix}\right). \tag{3.1}$$

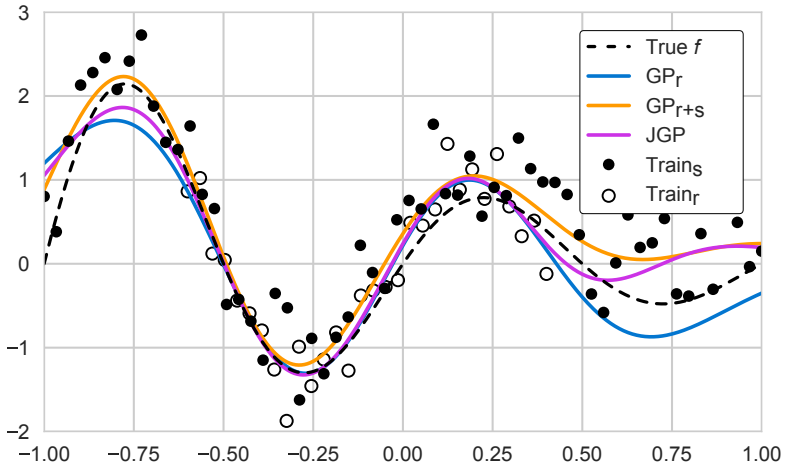


Figure 3.1: Toy experiment illustrating the Joint GP method compared to a GP using only the real data (GP_r) and a GP using the simulated data by pooling real and simulated data together GP_{r+s}. In the regions where there is no in-situ (real) training data, the GP_r tends to its mean function (0 in this case) and the GP_{r+s} assigns too much importance to the simulated data which is slightly biased with respect to the real data.

The resulting predictive mean and variance then takes the form

$$\mu_{\mathrm{JGP}}(\mathbf{x}_*) = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{V})^{-1} \mathbf{y}, \tag{3.2a}$$

$$\sigma^2_{\mathrm{JGP}}(\mathbf{x}_*) = c_* - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{V})^{-1} \mathbf{k}_*, \tag{3.2b}$$

$$\text{where } \mathbf{V} = \mathrm{diag}(\underbrace{1, \ldots, 1}_{r}, \underbrace{\gamma^{-1}, \ldots, \gamma^{-1}}_{s}).$$

When γ, which we shall call the *trust parameter* is low, and thus the lower diagonal of $\mathbf{V}$ high, the contribution to the predictive mean of the simulated data-points is minimized, and vice versa. We shall now discuss how to fit γ in such a way that it improves prediction on real data-points.

3.1.2 Hyperparameter optimization

Maximizing the likelihood of all data sources simultaneously, as done in the Gaussian process framework of Bonilla et al. (Bonilla et al., 2008), can lead to a phenomenon known as *negative transfer* (see, e.g., Rosenstein et al. (Rosenstein et al., 2005)). This means that the auxiliary data differs sufficiently from the data that one wishes to model, resulting in a poorer predictive model than would have resulted from not making use of it.

In order to avoid negative transfer, we only maximize the likelihood of the real data. We accomplish this by using the leave-one-out (LOO) likelihood which considers the likelihood of each data-point given the rest, thus incorporating the simulated data, but we only sum over the real data-points. This is reminiscent of the work in (Leen et al., 2012), who construct a focused *model*, whereas we in this work perform focused *inference*. This work is also related to multi-kernel learning (Melkumyan & Ramos, 2011) which attempts to add sufficient model flexibility to model each signal adequately and thus avoid negative transfer.

Leave-one-out likelihood:
The predictive probability of a single training data point conditioned on the remaining data is a normal distribution determined by Eq. (2.5), using all data points but the i'th. Thus, the predictive log-likelihood leaving out training point i can be expressed as

$$\log p(y_i|\mathbf{X}_{-i},\mathbf{y}_{-i},\boldsymbol{\theta}) = -\frac{1}{2}\log 2\pi\sigma_i^2 - \frac{(y_i-\mu_i)^2}{\sigma_i^2}.$$

From this we can construct the LOO likelihood by summing over each data point and fit the hyperparameters to maximize it. We modify this approach here, by only summing over the real data points

$$L_{\mathrm{LOO}}(\mathbf{X},\mathbf{y},\boldsymbol{\theta}) = \sum_{i=1}^{r} \log p(y_i|\mathbf{X}_{-i},\mathbf{y}_{-i},\boldsymbol{\theta}). \tag{3.3}$$

In computing r different predictive means and variances, it appears that we have to invert r slightly different covariance matrices. Luckily, there is a way around this computationally very inefficient approach, which involves simply computing the inverse of the complete covariance matrix (Sundarajan & Keerthi, 2001). Instead of using Eq. (2.5) a total of r times to evaluate the likelihood function, the following equations may be used:

$$\mu_i = y_i - \frac{[\mathbf{K}^{-1}\mathbf{y}]_i}{[\mathbf{K}^{-1}]_{ii}}\,, \quad \sigma_i^2 = \frac{1}{[\mathbf{K}^{-1}]_{ii}}\,, \tag{3.4}$$

where $[\cdot]_i$ denotes the i'th element of a vector, and $[\cdot]_{ii}$ is the i'th diagonal element of a matrix.

3.1.3 Improved prediction and extrapolation

The joint Gaussian process leverages data from the PROSAIL RTM described in Sec. 1.1.1. For details on the sampling scheme for the parameters used to generate the simulated data, see (Svendsen et al., 2017). Fitting the hyperparameters of the model as described above we are able to determine a measure of how much trust to put in the simulated data. This makes the incorporation of simulated data less likely to negatively affect the prediction on the real data. In (Svendsen et al., 2017) we use PROSAIL simulated data to aid the prediction of LAI from Landsat-8 spectra on rice fields. We show that for 10-fold cross validation, the JGP method either does not change the performance, or slightly improves it. This stands in contrast to the methods which only model simulated data, or which pool real and simulated data together (referred to as GP_s and GP_{r+s} respectively) and weigh the simulated data too high. The true advantage of the method, however, becomes evident when considering extrapolation cases, i.e. when the test data lies outside the region of the training data.

Table 3.1 shows the performance of the different approaches to using simulated data to improve regression including the baseline of only modelling the real data (GP_r). In order to simulate an extrapolation scenario we split the training and test data by the intensity of the green light band, so that the training data will contain little green light and the testing set will contain more green light. For each of the six datasets (collected from three sites

Table 3.1: Performance in terms of RMSE of the GP_r, GP_{r+s}, GP_s and JGP methods when dividing the real data so that test and training data are well-separated domains. The top and the bottom rows (seperated by bold horizontal line) hold results from the 50-50 and 75-25 partition schemes respectively.

	Dataset	GP_r	GP_{r+s}	GP_s	JGP
50-50	Spain 2015	3.78	1.26	1.28	3.12
	Greece 2015	3.05	1.30	2.41	2.28
	Italy 2015	1.82	1.63	1.50	1.56
	Spain 2016	4.31	1.65	1.50	2.92
	Greece 2016	2.90	1.78	1.87	2.69
	Italy 2016	1.91	1.36	1.70	0.77
75-25	Spain 2015	2.30	0.914	1.29	1.72
	Greece 2015	1.80	1.29	2.85	1.31
	Italy 2015	1.16	1.18	1.77	0.961
	Spain 2016	2.44	0.94	1.59	2.43
	Greece 2016	3.33	1.89	1.73	2.83
	Italy 2016	1.193	1.64	2.22	0.77

over two years) the JGP method is the only one which *consistently performs better than the baseline*. While other methods manage to improve more on the baseline than the JGP at times, they also perform worse than the baseline in other scenarios due to negative transfer.

3.2 Latent force models for soil moisture modelling

The method presented above shows how the physics knowledge encoded in simulated data can be used to improve the predictions of a GP model, especially in cases of extrapolation. In the following, we present a soil moisture (SM) prediction application of a GP model which is *derived directly from the governing equations* of the problem at hand. This implies strong model assumptions, but also results in model hyperparameters that have a clear physical interpretation, which is a strong advantage of this type of model.

In their paper about latent force models (LFMs) (Alvarez et al., 2009), Alvarez et al. assume that the observations in a set of time series are governed by an underlying ordinary differential equation (ODE) with a GP forcing. They show that the solution to this ODE is itself a Gaussian process over the outputs with a multi-output kernel which contains parameters of the underlying ODE. The following sections explain and derive the model as well as present an application to SM modelling, allowing us to reconstruct the precipitation.

3.2.1 First order ODE latent force model

A first order ODE is a useful model for soil moisture data dynamics (Delworth & Manabe, 1988), capturing the exponential decay behaviour exhibited in SM data[1]. Following

[1]Note that more sophisticated approaches exist which model SM losses in terms of drainage, runoff and evapotranspiration as a piecewise linear function of SM (Laio et al., 2001). This results in a non-linear differential equation, breaking the LFM model assumptions. Nevertheless, modelling the SM system as an ODE is a good approximation which proves effective and informative as seen in Sec. 3.2.2.

(Alvarez et al., 2009) we therefore consider an inhomogenous ODE of the form

$$\frac{\mathrm{d}y_q(t)}{\mathrm{d}t} + D_q y_q(t) = B_q + \sum_{r=1}^{R} S_{rq} f_r(t), \tag{3.5}$$

where f_r is a forcing on the system such as rain, adding moisture, or radiation, evaporating it. B_q is a parameter mainly related to soil hydraulic properties while D_q is the *decay rate*. The decay rate is the inverse of the so-called *e-folding time* which is typically used as a measure of soil moisture persistence. This ODE can be solved, e.g. as the convolution of the forcing with the Green's function of the ODE (Arfken & Weber, 1999), to obtain

$$y_q(t) = \frac{B_q}{D_q} + \sum_{r=1}^{R} L_{rq}[f_r](t) \tag{3.6}$$

where

$$L_{rq}[f_r](t) = S_{rq} \exp(-D_q t) \int_0^t f_r(\tau) \exp(D_q \tau) \mathrm{d}\tau$$

is a linear operator. Since applying a linear operator to a Gaussian process result in another Gaussian process $y_q(t)$ (Bishop, 2006), we can perform inference if we can find an expression for the corresponding kernel- or covariance-function

$$k_{y_p y_q}(t,t') = \mathrm{cov}(y_p(t), y_q(t')). \tag{3.7}$$

Without loss of generality let us assume that the number of latent forces is $R = 1$. Inserting (3.6) into (3.7), and using the properties of the covariance and the fact that integration is a linear operation we get

$$\begin{aligned} k_{y_p y_q}(t,t') &= S_p S_q \exp(-D_p t - D_q t') \times \\ &\quad \mathrm{Cov}\left(\int_0^t f_r(\tau) \exp(D_p \tau) \mathrm{d}\tau, \int_0^{t'} f_r(\tau') \exp(D_q \tau') \mathrm{d}\tau' \right) \\ &= S_p S_q \exp(-D_p t - D_q t') \times \\ &\quad \int_0^t \exp(D_p \tau) \int_0^{t'} \exp(D_q \tau') \mathrm{Cov}(f(\tau), f(\tau')) \mathrm{d}\tau' \mathrm{d}\tau. \end{aligned} \tag{3.8}$$

Whether an analytical solution is available to this double integral will depend on the covariance function which the latent function GP is assumed to have $\mathrm{Cov}(f(\tau), f(\tau')) = k_{ff}(\tau, \tau')$. Under the assumption that each latent force GP is independent and has an EQ covariance function, the resulting multi-output kernel does in fact have an analytical expression given by (Lawrence et al., 2007)

$$k_{y_p y_q}(t,t') = \sum_{r=1}^{R} \frac{S_{rp} S_{rq} \sqrt{\pi} l_r}{2} [h_{qp}(t',t) + h_{pq}(t,t')], \tag{3.9}$$

where

$$h_{pq}(t',t) = \frac{\exp(\nu_{rq})}{D_p + D_q}\exp(-D_q t')\Big\{\exp(D_q t) \times \left[\operatorname{erf}\left(\frac{t'-t}{l_r} - \nu_{rq}\right) + \operatorname{erf}\left(\frac{t}{l_r} + \nu_{rq}\right)\right] - \exp(-D_p t)\left[\operatorname{erf}\left(\frac{t'}{l_r} - \nu_{rq}\right) + \operatorname{erf}\left(\nu_{rq}\right)\right]\Big\} \tag{3.10}$$

in which $\operatorname{erf}(x)$ is the real valued error function and $\nu_{rq} = l_r D_q/2$.
With a multi-ouput kernel encoding the covariance between all the observations from different sources we can write up the log marginal likelihood of all the data, as described in 2.1, and optimize it with respect to kernel hyperparameters. Automatic differentiation is employed using the python package autograd[2] and optimized with the adam algorithm (Kingma & Ba, 2015). This is very interesting because it means we can perform inference about the parameters of the underlying system of differential equations and thus learn something about the decay rates of the different signals. Furthermore, we can also derive an expression for the cross-covariance between latent forces and the outputs:

$$k_{y_q f_r}(t,t') = \frac{S_{rq}\sqrt{\pi} l_r}{2}\exp(\nu_{rq}^2)\exp(-D_q(t-t')) \times \left[\operatorname{erf}\left(\frac{t'-t}{l_r} - \nu_{rq}\right) + \operatorname{erf}\left(\frac{t'}{l_r} + \nu_{rq}\right)\right]. \tag{3.11}$$

This means that we can perform prediction on and visualize the latent forces which, as we will see below, have a clear physical interpretation.

3.2.2 Modelling Soil Moisture with LFM

Soil moisture is a key hydrologic state variable, important to the understanding of various climatological and meteorological processes (Babaeian et al., 2019). The monitoring of SM has many applications, such as drought prediction, agricultural yield prediction, and forest-fire and flood prevention to name a few. We consider three satellites for SM estimation: The Soil Moisture Ocean Salinity (SMOS) from ESA, the Advanced SCATterometer (ASCAT) from EUMETSAT and the Advanced Microwave Scanning Radiometer 2 (AMSR2) from NASA. The measurements are taken over an in-situ SM network in Spain named REMEDHUS (17 in-situ sensors (Sanchez et al., 2012)).

We consider 6 years of SM-estimates from the three satellites, from July 2010 to July 2016. The AMSR2 satellite has only produced data from its launch date in May 18, 2012, as shown in Fig. 3.2. Multioutput GPs like the LFM learn the covariances not only between function values of the same signal, but also between signals. This makes them extremely useful for gap-filling (Alvarez et al., 2011). In Fig. 3.2 we see the fit of a LFM with 3 latent functions fitted to the above-mentioned dataset. We see that the first two years (pre-launch) of SM-estimates using AMSR2 data have been reconstructed, capturing the peaks also found in the other time-series, yet following the characteristics of the sensor aboard AMSR2. The in-situ SM measurements taken at the REMEDHUS site can be

[2]Code at https://github.com/HIPS/autograd

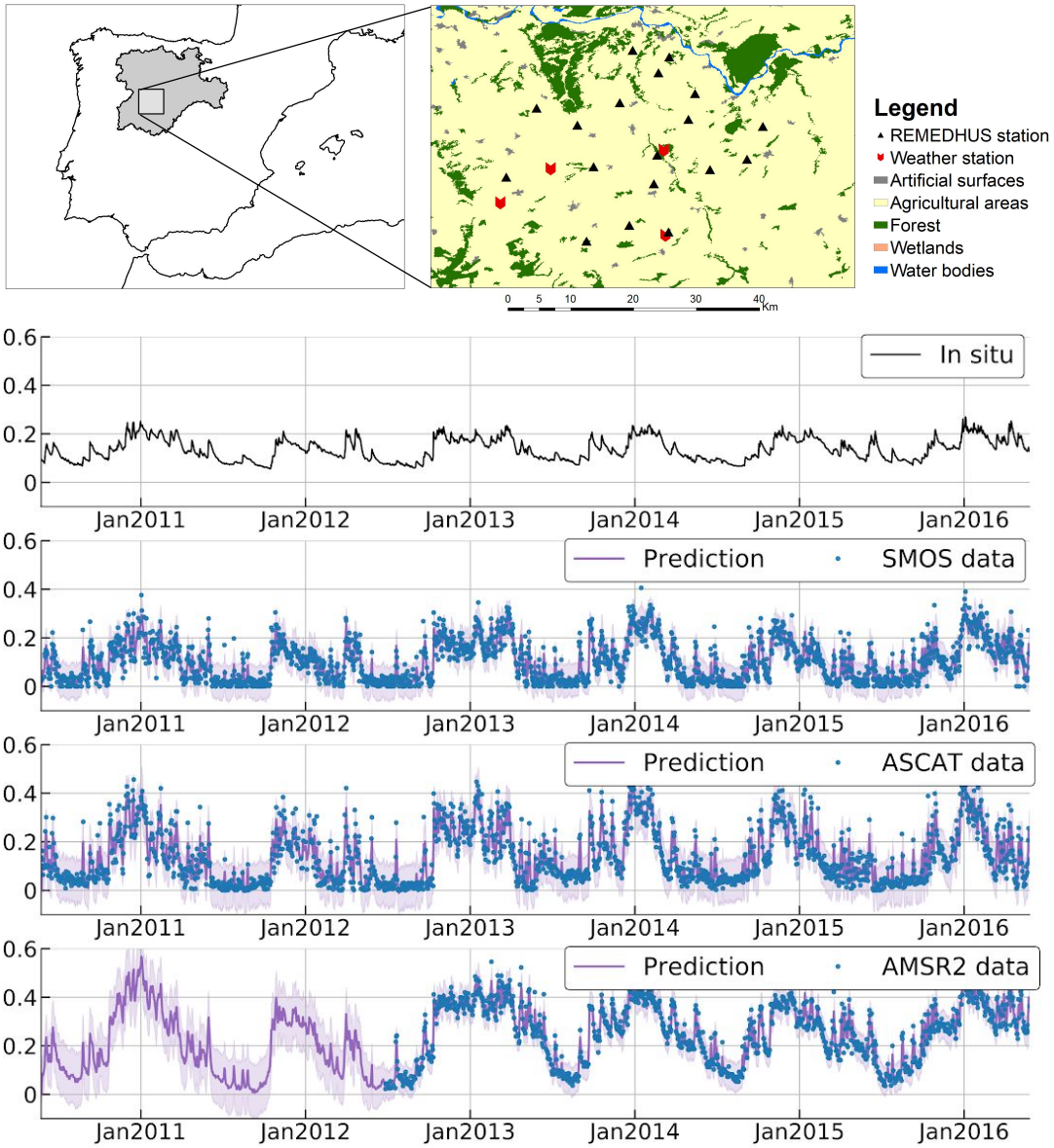


Figure 3.2: Results of application of an LFM to soil moisture time series at the REMEDHUS network using three latent forces. Top: layout of the 18 selected soil moisture stations and the 4 weather stations within the REMEDHUS validation site, located at the central part of the Duero river basin, Spain. Bottom: time series of *in-situ* (average of 18 stations), and satellite-based soil moisture estimates ($m^3 \cdot m^{-3}$) from SMOS, ASCAT and AMSR2 (blue dots denote the training data and purple lines and shaded regions represent the LFM predictions and confidence intervals).

used to assess the reconstruction AMSR2-estimates by comparing the correlation between in-situ SM measurements and SM-estimates derived from measurements by the SMOS and ASCAT satellites as well as the reconstructed AMSR2 time series:

Satellite	SMOS	ASCAT	Recon. AMSR2
R	0.867	0.828	0.868

The Pearson's correlation coefficient of the in-situ SM measurements with the reconstructed AMSR2 time series shows the effectiveness of this multi-output GP approach for

gap-filling.

The simplest mechanism of this system is that of water being added to the soil bringing SM up, after which it undergoes exponential decay, bringing it back down. The forcing, adding this moisture, is naturally that of precipitation, which matches very well with latent forces derived from fitting the LFM. Using the cross-covariance function between the observations and the (unobserved) latent forces in Eq. (3.11), we can reconstruct the latent forces. These are plotted in figure 3.4 along with the precipitation measured in-situ at the REMEDHUS site. We can apply the latent force for the classification problem of predicting rain-events. The area under curve (AUC) score for four different scenarios is shown in Fig. 3.3. The best prediction capabilities are obtained in terms of correlation coefficient and AUC when the three latent functions are considered in the LFM.

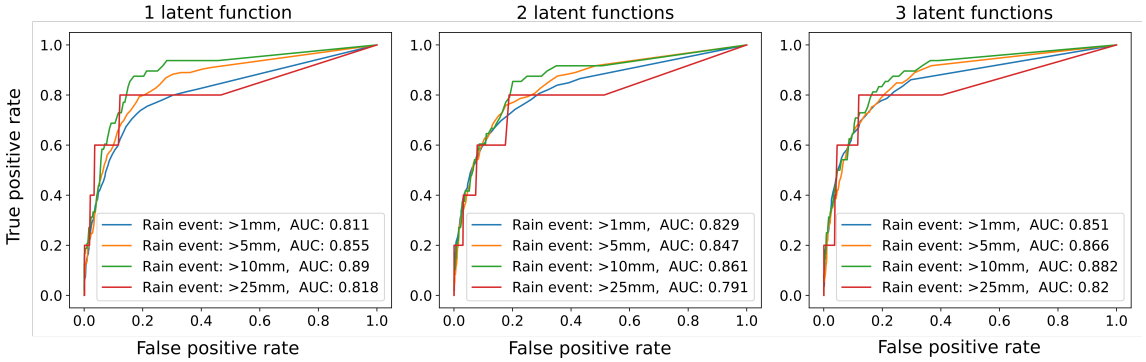


Figure 3.3: ROC curves for the classification of rain-events using the latent force which is most predictive of precipitation, for each of the three considered LFM models. Four scenarios were considered where a day was labeled as rainy if more than 1, 5, 10 and 25 mm of precipitation was measured. This corresponds to 417, 145, 48 and 5 rainy days respectively in the 6-year study period. We see that the classification performance for rain-events of higher than $1mm{\cdot}day^{-1}$, as measured by area under curve (AUC), increases with model complexity.

Table 3.2 shows the performance of the three models that use 1, 2 and 3 latent forces respectively. We see that, for model using 3 latent forces, the fitted e-folding time (inverse decay rate) fits better with the literature (Piles et al., 2018) and a latent force is found that matches the true percipitation better both in terms of correlation coefficient and AUC.

Precipitation estimation is important to the development of a proper understanding of the hydrologic cycle as water passes through the ocean, land, and atmosphere. It is therefore very interesting that the latent forces obtained from fitting an LFM to the soil moisture data correlate well with the in-situ precipitation. The LFM has different local minima which makes it difficult to find the exact coupling constant of the latent forces. Therefore, a scaling has been applied in the figures 3.4 to illustrate the correlation better. We remove negative values of the latent forces, as done in publications that attempt to model the precipitation in more direct ways (Brocca et al., 2013).

Figure 3.4 shows the first latent force which exhibits peaks that match well with the in-situ precipitation but has a lengthscale that varies too slowly to capture the spiky nature of the actual rainfall. The second latent force, however, captures the narrow peaks of the precipitation but also suffers more false positives, i.e. claims that moisture was added to the ground when no rain was recorded. The third latent force shown in Fig. 3.4 (bottom) captures the general trend of the soil moisture. Using three latent forces allows the model

	1 latent function			
	σ	τ	R	AUC
SMOS	1.14×10^{-3}	67.18	0.517	0.811
ASCAT	3.61×10^{-3}	69.54		
AMSR2	1.40×10^{-3}	125.71		
	2 latent function			
	σ	τ	R	AUC
SMOS	1.15×10^{-3}	23.27	0.521	0.829
ASCAT	1.51×10^{-3}	19.79		
AMSR2	0.44×10^{-3}	47.79		
	3 latent function			
	σ	τ	R	AUC
SMOS	0.89×10^{-3}	16.21	0.549	0.851
ASCAT	1.26×10^{-3}	11.04		
AMSR2	0.51×10^{-3}	16.27		

Table 3.2: Results of application of LFM models to satellite-based soil moisture time series over the REMEDHUS network using 1, 2, and 3 latent forces. The estimated input noise σ and e-folding time τ (days) obtained per each satellite are reported. The latent force which is more predictive of precipitation in each case is used to calculate: i) the Pearson correlation R of the obtained LF with in-situ precipitation measurements, and ii) the area under the curve (AUC) performance metric for classification of rain-events, in which a measured in-situ precipitation higher than 1 mm is considered a rain event (see Fig. 3.3 for more results).

to fit the data better in terms of marginal likelihood, but the third latent force cannot be interpreted as an actual physical phenomenon.

3.3 Concluding remarks

In this chapter we presented the GP framework developed in (Svendsen et al., 2017) which jointly models in-situ data and RTM-simulated data. By using LOO likelihood maximization over the in-situ data only, we arrive at a predicive model which does not overfit the simulated data. This allows us to safely incorporate simulated data when in-situ measurements are scarce. We saw that this framework was particularly good for extrapolation. As explained in (Svendsen et al., 2017), the JGP can be derived from a kernel ridge regression perspective, where the simulated data appear in an extra mean squared error-term in the loss function. In that sense, it is similar to the physics-guided regression methods which rely on physics based loss-functions (Karpatne et al., 2017). It is also worth noting that the JGP framework can be extended to multiple data sources, as described in (Svendsen et al., 2017), fitting a trust-parameter for each source.

Secondly, following the work of Alvarez et al. (Alvarez et al., 2009), we saw that one can derive a GP from an ODE by assuming a GP forcing on the differential equation system. This lead to a physically interpretable latent force which matched the (independently measured) rainfall at the site. Having learned the covariances between signals, the method also proved useful for gap-filling.

While including simulated data when performing regression on in-situ data lead to improved performance, it does not necessarily yield much insight, apart from an idea of

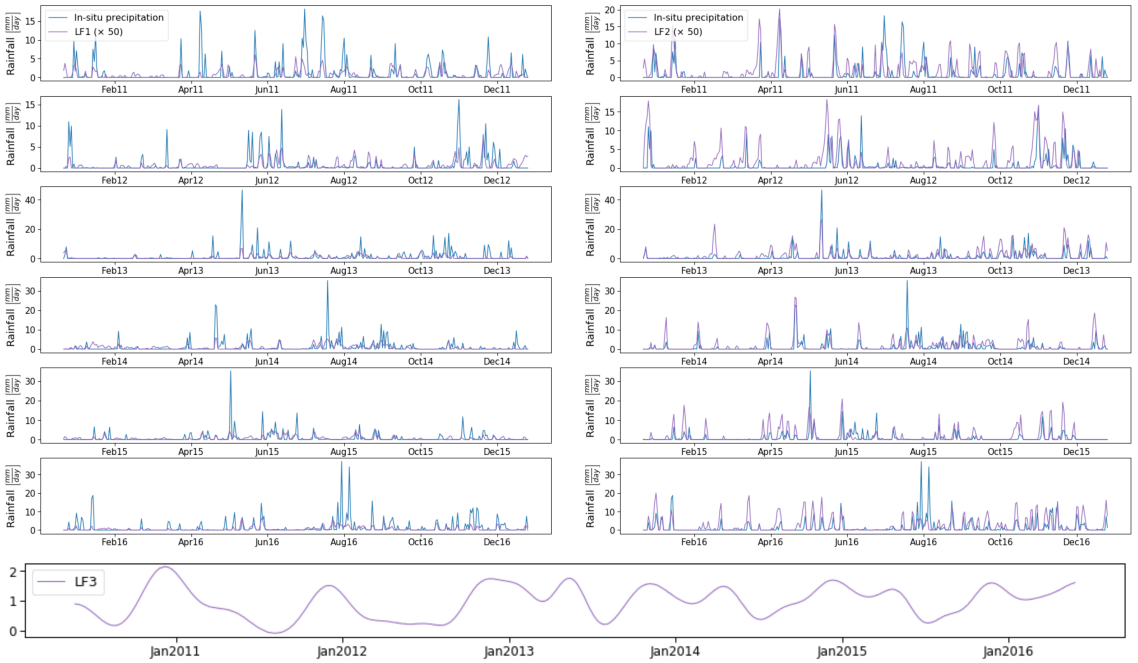
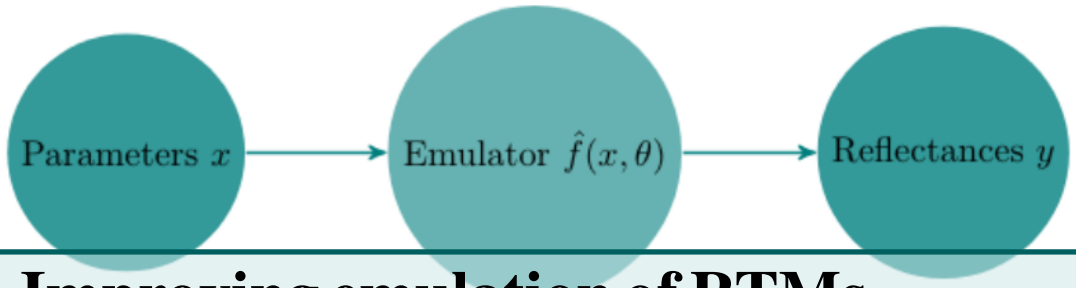


Figure 3.4: Inferred latent forces associated to the satellite soil moisture time series over REMEDHUS. Top: first (left) and second (right) estimated latent forces, plotted alongside in-situ precipitation measurements (mm·day^{-1}). To better illustrate the comparison, the negative values of the LFs are set to zero and a scaling factor is applied. Bottom: third estimated latent force, which may be related to the annual trend or seasonality inherent to Earth system processes.

how much to trust the simulated data. On the other hand, building an ML model based on governing equations of the system at hand, can lead to physics insight. The downside is that the assumptions of the model can make the it less flexible. The relevant example here, is that if gap-filling was our only goal, we could have explored a wide range of more flexible multioutput GP models (Alvarez et al., 2011).



4. Improving emulation of RTMs with active learning

Many scientific fields make use of computer code simulations to analyze systems of interest. Such *simulators* act as convenient approximations to reality, allowing us to study how diseases spread across a population, how force distributes on a load-carrying beam or how light interacts with the atmosphere of the Earth to name a few applications. There are, however, two important limitations associated with simulators:

- **Computational cost:** In an attempt to capture the true mechanics of the system of interest, numerical implementations of the relevant governing equations can become computationally cumbersome. This is undesirable as it hampers the ability to perform exhaustive simulations and sensitivity analysis (Sobol, 1993).
- **Mathematical tractability:** Computer codes often rely on decades of iterative development making use of various heuristics that improve accuracy but make the models less mathematically tractable and transparent. It is especially useful to be able to access an estimate of the derivative of the model. This is necessary for studying the propagation of uncertainty through the model, but is also essential when training machine learning methods that incorporate physical simulators in their likelihood function, such as the method presented in Sec. 5.

It is possible to obtain a computationally efficient and differentiable *emulator* of a computer code by generating a representative set of simulated input-output data pairs and using it to train a machine learning model. The question of how to generate such a dataset while running the simulator as few times as possible is addressed in this chapter.

This chapter is partly based on the publication:

1. **Svendsen, D.H.**, Martino, L. and Camps-Valls, G., 2020. Active emulation of computer codes with Gaussian processes–Application to remote sensing. Pattern Recognition, 100, p.107103.

4.1 Active multi-output Gaussian process emulator

In order to build a computationally efficient and differentiable emulator, we need a dataset of simulated input-output pairs generated by the computer code of interest. Computer codes are often slow to run, thus we want to evaluate them as few times as possible. Furthermore, the computational cost of some regression algorithms such as GPs, which are popular for emulation due to their uncertainty estimates (O'Hagan, 2006), rise with the number of training points. It is therefore important to choose a small, yet representative set of input-points at which to evaluate the computer code in question. A common approach to this problem is the Latin Hypercube sampling (LHS) scheme (Audze, 1977) or simply performing random sampling according to a physically meaningful distribution. These approaches, however, do not take into account the knowledge of the function behaviour that can be learned from fitting a regression model to the simulated data. In (Svendsen et al., 2020a), we provide an overview of the different methods developed in order to solve the sampling problem. We present an active learning (AL) (Settles, 2009) algorithm for solving said problem. Other approaches for designing adaptive RTM emulators can be found in (Vicent et al., 2019, 2020).

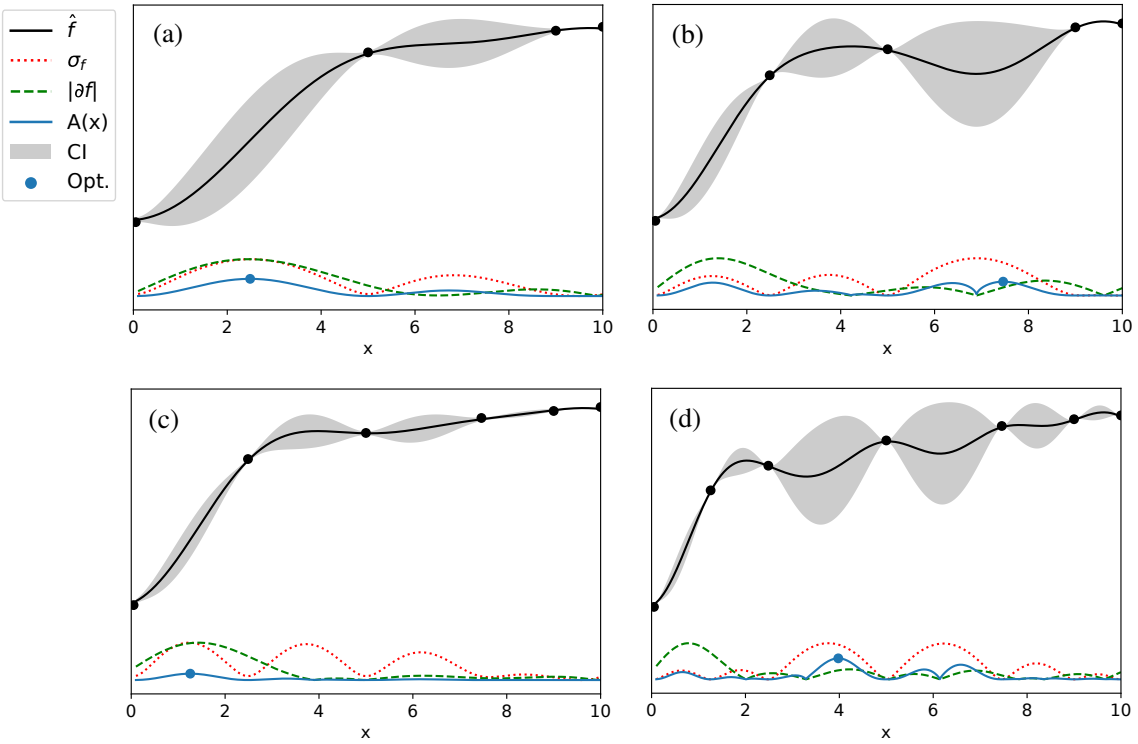


Figure 4.1: The presented method optimizes the selection of the most informative points (selected points are shown as black dots) to approximate an arbitrary multidimensional function iteratively. The example shows the first four iterations in a one-dimensional scenario. Starting from 4 points, a GP interpolator is built from which valuable information is derived (the predictive variance -green- and the gradient -red-) and then combined in an acquisition function (blue) that proposes the next point to sample (blue dot). The acquisition function admits many general forms and trades off geometry and diversity terms to account for attractiveness in the sample space.

4.1.1 Sequential and non-sequential sampling

It is important to remark that the emulation procedure presented in this work is intrinsically a sequential technique. This means that the method chooses an optimal point based on the already existing dataset of points, according to some criterion. Non-sequential techniques such as LHS, on the other hand, attempt to provide a set of input-points that cover in input-space in which to run the simulator. This implies knowing the number of training points needed in order to construct a good emulator beforehand. Should one decide to add a point to the training-data, the locations of the remaining points would have to be re-calculated.

This scenario can be avoided by sequentially adding simulated data-points. We also show that by using information learned from the simulated data, fewer evaluations of the potentially complex simulator are needed to construct an accurate emulator. In this sense, the emulation procedure is a parsimonious technique that applies, at each iteration, all previously obtained information about the underlying function.

4.1.2 Products of the algorithm

The active emulation procedure proposed in this work is a methodology that delivers: (a) an accurate GP emulator while evaluating the computer code as little as possible, (b) a final set of data-pairs $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m_t}$ as a Lookup Table (LUT; other regression methods can be applied using the obtained set of points), and (c) useful statistical information about the model $\mathbf{f}$, such as predictive variance and gradients of the learned function, which can be further used for model inversion and error propagation analyses.

4.1.3 General framework

Consider a D-dimensional bounded input space $\mathcal{X}$, i.e., $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$. Furthermore, let $\mathbf{f}(\mathbf{x})\colon \mathcal{X} \to \mathbb{R}^P$ denote a complex system with P outputs. Finally, $t \in \mathbb{N}$ denotes the index of the AL algorithm, and m_t the number of data points $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m_t}$ used by the algorithm at iteration t, where

$$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k), \tag{4.1}$$

$\mathbf{y}_k = [y_{1,k}, \ldots, y_{P,k}]^\intercal$ and $k = 1, \ldots, m_t$. At each iteration t, given the data points $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m_t}$, the AL method constructs regression function $\widehat{\mathbf{f}}_t(\mathbf{x})$. Note that we are treating the forward problem and thus the regression function maps vectors of physical parameters into reflectances. Based on this function, an acquisition function $A_t(\mathbf{x})\colon \mathcal{X} \to \mathbb{R}$ is made in order to suggest which regions of the space require additional data points. That is, an optimization step is performed for obtaining the next input $\mathbf{x}_{m_t+1}$:

$$\mathbf{x}_{m_t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} A_t(\mathbf{x}). \tag{4.2}$$

Then, we compute $\mathbf{y}_{m_t+1} = \mathbf{f}(\mathbf{x}_{m_t+1})$ and add this datapair to the dataset. The acquisition function reflects the knowledge contained in the regression function about where in the input space it would be most beneficial to evaluate the simulator code to obtain more information.

In (Svendsen et al., 2020a) we argue that $A_t(\mathbf{x})$ should consist of two terms: One *diversity* term taking high values in regions of the input space where data is scarce, and

Table 4.1: Generic Active Emulator.

1. Set $t = 0$, select initial points $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m_0}$, and maximum number of simulated data M.
2. While $m_t < M$:
   (a) Given $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m_t}$, build function $\widehat{\mathbf{f}}_t(\mathbf{x})$.
   (b) Build the acquisition function $A_t(\mathbf{x})$ from $\widehat{\mathbf{f}}_t$, and obtain the new input
   $$\mathbf{x}_{m_t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} A_t(\mathbf{x}). \tag{4.4}$$
   (c) Obtain outputs $\mathbf{y}_{m_t+1} = \mathbf{f}(\mathbf{x}_{m_t+1})$.
   (d) Update dataset $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m_t+1}$.
   (e) Set $m_{t+1} = m_t + 1$ and $t \leftarrow t + 1$.
3. Build the interpolating function $\widehat{\mathbf{f}}_t(\mathbf{x})$.
4. Return final set of optimal nodes $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m_t}$ as a Look-up Table (LUT), as well as the gradient and the predictive variance of the predictive model $\widehat{\mathbf{f}}_t(\mathbf{x})$.

one *geometric* term which takes high values in regions of high variability of the function $\mathbf{f}(\mathbf{x})$. We consider acquisition functions obtained by the multiplication of the geometry term $G_t(\mathbf{x})$ and the diversity term $D_t(\mathbf{x})$, i.e. functions of the form:

$$A_t(\mathbf{x}) \quad = \quad G_t(\mathbf{x})D_t(\mathbf{x}). \tag{4.3}$$

We found that $A_t(\mathbf{x})$ has many local maxima, and thus using pure gradient-based optimization is not optimal. We start optimization with a random search and perform gradient ascent initialized in the best candidate point. We refer to this active learning approach for constructing emulators as *active emulation*, and an overview of the generic algorithm is given in Table 4.1. Figure 4.1 shows an illustrative example of the building blocks of the active emulation methodology presented.

4.1.4 Specific implementation

The section above raises the question of what forms the diversity and geometry terms take. In order to answer this we must first select a regression algorithm, which in this work will be a multi-output Gaussian process (MOGP) method. There are many MOGP methods to choose from (for a review, see (Alvarez et al., 2011)) and any of them could be applied with our active learning algorithm, but in (Svendsen et al., 2020a) we make use of a particularly simple scheme of fitting P single-output GPs.

Diversity term

The predictive distribution of a GP is defined by its predictive mean and variance (see Eq. (2.5)). The variance at a given input vector $\mathbf{x}^*$ is an expression of the predictive uncertainty and is a function of how different (as measured by the kernel function) $\mathbf{x}^*$ is from the input vectors of the training dataset. It is therefore a useful metric for *diversity*, taking on large values in regions of $\mathcal{X}$ where training data is scarce. Denoting the MOGP predictive variance at iteration t corresponding to the p'th output as $\sigma^2_{p,t}(\mathbf{x})$, we define the diversity

term as

$$D_t(\mathbf{x}) := \sigma^2_{1,t}(\mathbf{x}) \odot \sigma^2_{2,t}(\mathbf{x}) \odot \sigma^2_{3,t}(\mathbf{x}) \ldots \odot \sigma^2_{P,t}(\mathbf{x}), \tag{4.5}$$

where $\odot$ represents a generic mathematical operation such as sum $(+)$ or multiplication $(\times)$.

Geometry term

With the right choice of kernel function, the GP predictive mean is differentiable. This allows us to define the function that computes the L2-norm of the gradient vector, corresponding to output p at iteration t:

$$\mathtt{Gr}_{p,t}(\mathbf{x}) = \left\| \nabla_x \widehat{f}_t(\mathbf{x}) \right\|. \tag{4.6}$$

This is descriptive of how much variability the underlying function is exhibiting at a given $\mathbf{x}^*$. Since regions of high function variability are more difficult for the regression algorithm to successfully represent, and thus require more training points, we use the following geometry term:

$$G_t(\mathbf{x}) := \mathtt{Gr}_{1,t}(\mathbf{x}) \odot \mathtt{Gr}_{2,t}(\mathbf{x}) \odot \mathtt{Gr}_{3,t}(\mathbf{x}) \ldots \odot \mathtt{Gr}_{P,t}(\mathbf{x}). \tag{4.7}$$

With these building blocks, it is possible to construct different types of acquisition functions which emphasize data sampling in regions of data scarcity and high function variability. Table 4.2 shows the different combinations of diversity and geometry terms used in the experimental section.

Table 4.2: Acquisition functions for a multi-output emulator and their shorthand notation used in the experimental section.

$A_t(\mathbf{x})$	Shorthand
$\sum_{p=1}^{P} \sigma^2_{p,t}(\mathbf{x})$	ΣD
$\prod_{p=1}^{P} \sigma^2_{p,t}(\mathbf{x})$	ΠD
$\sum_{p=1}^{P} \sigma^2_{p,t}(\mathbf{x}) \sum_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΣD$\times\Sigma$G
$\sum_{p=1}^{P} \sigma^2_{p,t}(\mathbf{x}) \prod_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΣD$\times\Pi$G
$\prod_{p=1}^{P} \sigma^2_{p,t}(\mathbf{x}) \sum_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΠD$\times\Sigma$G
$\prod_{p=1}^{P} \sigma^2_{p,t}(\mathbf{x}) \prod_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΠD$\times\Pi$G

4.2 Experimental results

This section presents experimental results of the our AE framework in synthetic and real (Earth-observation) systems. The described active MOGP emulation (AMOGAPE)[1]

[1] Code available at https://github.com/dhsvendsen/AMOGAPE

method is compared to standard algorithms in the literature, namely random exploration/sampling and most notably Sobol's sampling (Bratley & Fox, 1988) and the Latin Hypercube Sampling method (McKay et al., 1979). Algorithms are compared in terms of accuracy and convergence rates in problems of different input and output dimensionality.

4.2.1 Toy Experiment: Unidimensional multi-output emulation

We consider a multi-output toy example with scalar inputs $x \in \mathbb{R}$ where we can easily compare the achieved approximation $\widehat{\mathbf{f}}_t(x)$ with the underlying function $\mathbf{f}(x)$. In this way, we can exactly check the true accuracy of the obtained approximation using different schemes. For the sake of simplicity, we consider the following multi-output mapping

$$\mathbf{f}(x) = [\log(x), 0.5\log(3x)], \quad x \in (0, 10], \tag{4.8}$$

then $D = 1$ and $P = 2$ (two outputs). Even in this simple scenario, the procedure used for selecting new points is relevant. We start with $m_0 = 4$ support points, $\mathbf{X}_0 = [0.1, 3.4, 6.7, 10]$, apply an independent GP per output, and for AMOGAPE we use the acquisition function denoted as $\Pi D \times \Pi G$ in Table 4.2.

Comparison among sequential methods

It is important to remark that the active emulators presented in this work are intrinsically sequential techniques. This means that the nodes in $\mathbf{X}_{t-1}$ all always contained in $\mathbf{X}_t$, where $\mathbf{X}_t$ is the matrix of input-points at iteration t. Therefore, for a fair comparison we have to consider other sequential algorithms. We sequentially add 20 additional points to $\mathbf{X}_t$, using different sampling strategies: AMOGAPE, uniform points randomly generated in $(0, 10]$, a sequential Sobol sequence, and a sequential version of the Latin Hypercube Sampling procedure (Seq-LHS). Seq-LHS simply generates 20 nodes following the LHS procedure and then adds one to $\mathbf{X}_t$ at each iteration (without replacement). Note that, at each run, the results can vary even for the deterministic procedure due to the optimization of the hyperparameters. We average all the results over 500 independent runs. For model comparison, we compute the root mean square error (RMSE) between $\widehat{\mathbf{f}}_t(x)$ and $\mathbf{f}(x)$ at each iteration, and show the evolution of the (averaged) RMSE versus the number of support points m_t (that is $m_t = t + m_0$) in Figure 4.2. We can observe that the AMOGAPE scheme outperforms the other methods, providing the smallest RMSEs between $\mathbf{f}(x)$ and $\widehat{\mathbf{f}}_t(x)$.

Comparison with non-sequential methods

In order to provide an exhaustive numerical analysis we also compare AMOGAPE with non-sequential techniques where the input matrix $\mathbf{X}_t$ can be completely different from $\mathbf{X}_{t-1}$ (whereas, in AMOGAPE, the nodes in $\mathbf{X}_{t-1}$ all always contained in $\mathbf{X}_t$). This approach would not be used in practice, but serves as an interesting comparison of AMOGAPE with one-shot space-filling algorithms. More specifically, we consider:

- Deterministic grid: at each step, we consider an equal-spaced set of points (deterministically chosen). Clearly, at each step, all the points in $\mathbf{X}_{t-1}$ are not considered, replaced by new nodes.
- Standard LHS: also in this case, at each iteration all the previous points are changed.

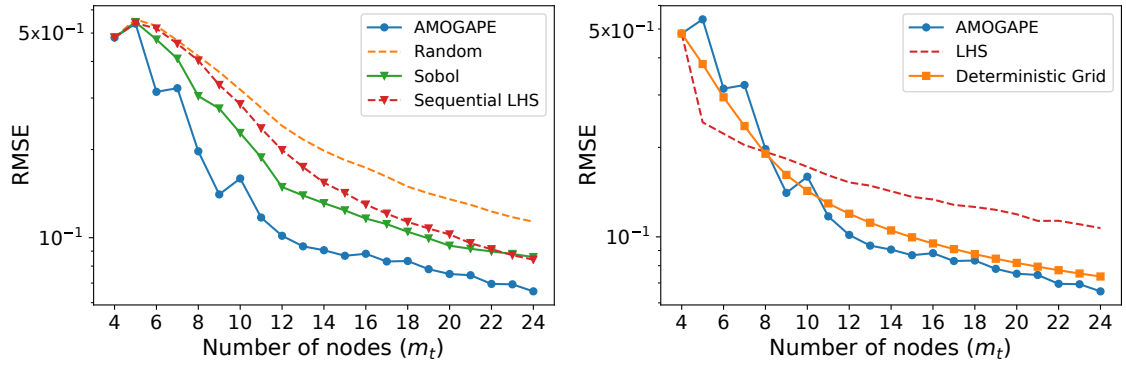


Figure 4.2: RMSE (in log-scale) between $\mathbf{f}(x)$ and $\widehat{\mathbf{f}}_t(x)$ versus the number of nodes m_t, that is $m_t = t+4$ in this example ($D = 1$ and $P = 2$). **(a)** Comparison with sequential methods (i.e., fair comparison, with the same computational cost). **(b)** Comparison with two non-sequential methods (number of evaluation of $\mathbf{f}(x)$ is $\sum_{t=1}^{T} m_t = \frac{m_T(m_T+1)}{2}$), and AMOGAPE (number of evaluation of $\mathbf{f}(x)$ is m_T).

Clearly, these two schemes at each iteration evaluate the underlying function in m_t new nodes. Therefore, they are more costly than AMOGAPE. The total number of evaluations of $\mathbf{f}(x)$ for AMOGAPE is m_T whereas, for the non-sequential schemes above, is $\sum_{t=1}^{T} m_t = (m_T^2 + m_T)/2$. However, even in this unfair comparison for our method, Figure 4.2(b) shows that AMOGAPE is able to provide the smallest error when more than 12 new points are incorporated. This illustrates that the gradient term encoded in the AMOGAPE adds useful information to the active learning scheme.

4.2.2 Application to remote sensing: Emulating a radiative transfer model

We now apply the AMOGAPE algorithm to the PROSAIL radiative transfer model. We vary two of the most important input parameters, namely leaf area index (LAI) and chlorophyll content (Chl)[2], and project the output using the Landsat-8 spectral response function. Seeing as this satellite has 9 bands, this problem has input and output dimensionalities $D = 2$ and $P = 9$ respectively. For details on what values we set for the remaining parameters, see (Svendsen et al., 2020a).

We compare the different acquisition functions shown in Table 4.2 with the random sampling strategy, using a truncated Gaussian $\mathcal{N}_{\mathcal{T}}$(Chl,LAI) - a physically meaningful prior used in the work of Campos et al. (Campos-Taberner et al., 2016). Each method starts with $m_0 = 30$ points sampled from the same prior and updates according to active emulation algorithm, except the random sampling method. For each iteration, the performance of the emulator trained on the simulated dataset generated by each algorithm is shown in Fig. 4.3. The performance is measured in terms of RMSE, averaged over the 9 output dimensions, on a test dataset of 5×10^4 points. The results are averaged over 15 runs.

We see that it is possible to perform better using the AMOGAPE approach on our test-set than by sampling randomly from $\mathcal{N}_{\mathcal{T}}$(Chl,LAI). It is interesting to note that methods using ΣD$\times\Sigma$G and ΣD perform similarly, implying that the ΣD term is governing the acquisition function. Similarly, methods using ΠD and ΠD$\times\Sigma$G perform equally well, showing that ΠD is the most influential term. The acquisition function ΣD$\times\Pi$G, which

[2]The remaining input parameters are kept constant as explained in (Svendsen et al., 2020a).

penalizes a zero-gradient in any of the output dimension, relies too much on geometric information and performs the worst. It seems that the information source which is included in product form governs $A(\mathbf{x})$. The ΠD×ΠG method manages to strike a balance between the two sources of information. All in all, the best performing methods are ΣD and ΣD×ΣG. This hints at the idea that the product form is too restrictive, i.e. considering a point uninteresting if the predictive variance is close to zero in only one of the output-dimensions.

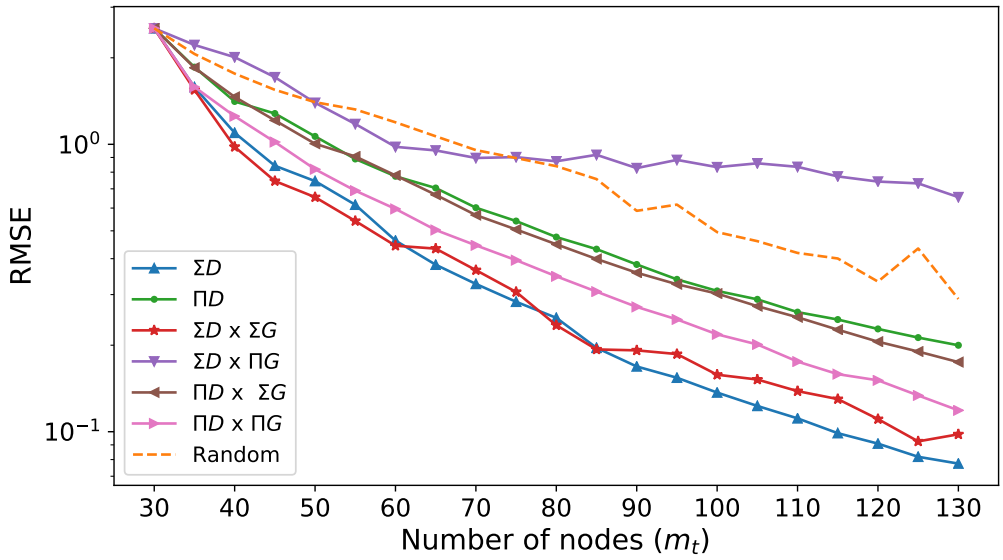


Figure 4.3: Function approximation errors by different acquisition functions, cf. Table 4.2, and for different numbers of selected nodes m_t in a bidimensional PROSAIL problem. Only the best performing acquisition functions are compared here to random sampling.

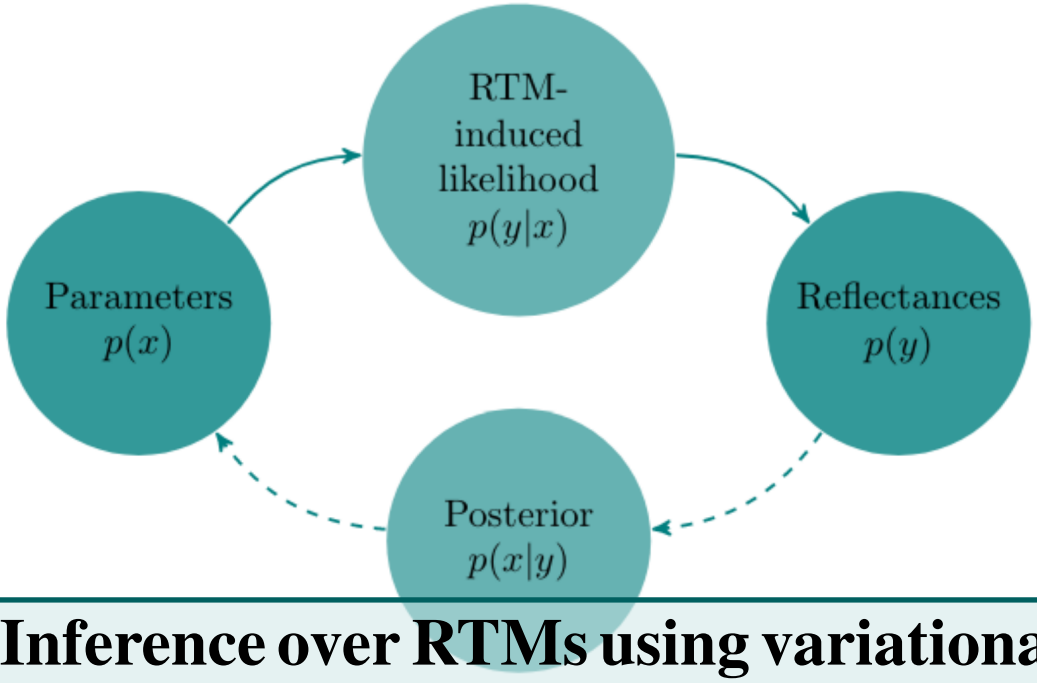
In (Svendsen et al., 2020a), more exhaustive experiments are carried out on toy data, as well as on a 3-dimensional PROSAIL problem.

4.3 Concluding remarks

Simulators are often computationally expensive and mathematically opaque. Through emulation it is possible to speed up computation greatly and to evaluate gradients and higher order derivatives of the model. In order to build an emulator, one needs to evaluate the simulator in a set of inputs to build a training dataset, especially in regions where the behaviour of the simulator is complex. It can be tempting to generate a very dense grid of simulated data in order to capture all such regions, but that is often prohibitively expensive. We have shown that we can use what the regression model learns about the gradient of a radiative transfer model, combined with the predictive variance, to generate an efficient active learning scheme. This results in an accurate emulator which requires fewer evaluations of the RTM.

While the integration of computer codes such as RTMs in a machine learning cost function is an effective way to penalize deviation from physically meaningful predictions, it can be hard to implement if the physical model is not analytically tractable. Apart from penalizing the deviation of a regression model f from the true output: $\|f(\mathbf{x}) - y\|$ one can

discourage the deviation from a prediction of a physical model $\phi : \mathcal{X} \to \mathcal{Y}$ evaluated in the sample input: $\|f(\mathbf{x}) - \phi(\mathbf{x})\|$ as done in (Karpatne et al., 2017). One might, however, wish to pass the prediction through a physical model that maps in the opposite direction of the regression $\psi : \mathcal{Y} \to \mathcal{X}$ and compare with the original input: $\|\psi(f(\mathbf{x})) - \mathbf{x}\|$. This requires access to the gradient ψ, which can be obtained with an emulator. In this way emulation enables other ML algorithms to incorporate physics knowledge.



5. Inference over RTMs using variational and expectation maximization methods

Radiative transfer models encode the *forward direction* of the parameter retrieval problem that we have been studying. We have seen how such a forward model is useful for direct inversion by generating simulated training data to train a regression algorithm for mapping in the inverse direction. A forward model can also, however, be used to define a likelihood model of a probabilistic approach. In other words, given some vector of physical parameters $\mathbf{x}$ (atmospheric or canopy properties), the forward RTM model induces a likelihood function $p(\mathbf{y}|\mathbf{x})$, which links physical parameters with the observed reflectances $\mathbf{y}$. This interpretation opens up new options for inference of, e.g., a prior distribution over physical parameters.

In this chapter we address a general problem: Learning the distribution of the physical parameters, instead of only providing a point-wise estimation of these parameters. Provided a dataset of observed reflectances $\mathbf{y}'$, our goal is twofold: learning the marginal density $p(\mathbf{x})$ and obtaining an approximation of the posterior distribution $p(\mathbf{x}|\mathbf{y}')$. Note that the posterior represents a probabilistic inverse model, i.e., given $\mathbf{y}'$ we can obtain a prediction of the causes $\mathbf{x}$ and related uncertainty measures. Here, we propose and compare two different approaches which allows us to infer physical parameters using an RTM forward model. One approach is based on Variational Autoencoders (VAEs) (Kingma & Welling, 2013) and the other is based on Monte Carlo expectation maximization (MCEM) (Wei & Tanner, 1990). We will show that each approach has different pros and cons. While the MCEM approach is mathematically elegant and flexible and has good convergence properties, its application in practice is computationally demanding. On the other hand the proposal based on VAEs obtains good results and is fast, yet it is not able to describe multimodal distributions. We illustrate these properties in several toy examples of varying sample sizes and complexity, as well as with the PROSAIL RTM.

This chapter is largely based on the following scientific paper:

- **Svendsen, D.H.**, Hernández-Lobato, D., Martino, L., Laparra, V. and Camps-Valls, G. "Inference over Radiative Transfer Models using Variational and Expectation Maximization Methods". (submitted)

5.1 Problem setting

Notationally, we consider the reflectance vector $\mathbf{y} \in \mathbb{R}^P$ and physical parameter vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$. An RTM model represents the underlying mapping from $\mathbf{x}$ to $\mathbf{y}$, that we denote as $\mathbf{f}(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^P$. The complete observation model is given by

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(\mathbf{e}|\mathbf{0}, \sigma^2\mathbf{I}), \tag{5.1}$$

where $\mathbf{I}$ is a unit $P \times P$ matrix. The observation model defines the likelihood function as

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{f}(\mathbf{x}), \sigma^2\mathbf{I}). \tag{5.2}$$

Note that by fixing $\mathbf{x}$, the conditional probability $p(\mathbf{y}|\mathbf{x})$ is Gaussian, but as a function of $\mathbf{x}$ the likelihood is a highly non-linear function due to the dependence on the RTM with the causes, i.e. $\mathbf{f}(\mathbf{x})$. We assume a Gaussian prior over $\mathbf{x}$'s,

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{S}), \tag{5.3}$$

where $\mathbf{m} \in \mathbb{R}^D$ and the $D \times D$ covariance matrix $\mathbf{S}$ are considered unknown. The posterior density given the observed data $\mathbf{y}$ over the causes can be expressed as

$$p(\mathbf{x}|\mathbf{y}) \quad \propto \quad p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{f}(\mathbf{x}), \sigma^2\mathbf{I})\mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{S}) \tag{5.4}$$

Our goals are: (a) To learn the prior parameters, vector $\mathbf{m}$ and matrix $\mathbf{S}$, and (b) to obtain an approximation of the posterior $p(\mathbf{x}|\mathbf{y})$, which serves as an inverse probabilistic mapping from $\mathbf{y}$ to $\mathbf{x}$. We assume that some set of data $\mathbf{y}$ is given. The two main ways of approaching this problem are a variational inference (VI) scheme on the one hand, and an expectation maximization method on the other.

5.2 Variational inference method

As described in Sec. 2.2.2, the idea of variational inference is to optimize the parameters of a *variational posterior* in order to come as close as possible to the *true posterior*. Following the approach of Kingma and Welling (Kingma & Welling, 2013) we choose a Gaussian variational posterior,

$$q(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathrm{NN}}(\mathbf{y}), \boldsymbol{\Sigma}_{\mathrm{NN}}(\mathbf{y})), \tag{5.5}$$

where $\boldsymbol{\mu}_{\mathrm{NN}}(\mathbf{y})$ and $\boldsymbol{\Sigma}_{\mathrm{NN}}(\mathbf{y})$ are parametrized by a Neural Network (NN) with parameters $\boldsymbol{\phi}$. These parameters $\boldsymbol{\phi}$ are the variational parameters of this problem. As in the variational inference scheme for DGPs, we tune the parameters, both those of the neural network and those of the prior $\{\boldsymbol{\phi}, \mathbf{m}, \mathbf{S}\}$, by maximizing the ELBO:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log \frac{p(\mathbf{x},\mathbf{y})}{q(\mathbf{x}|\mathbf{y})}\right] = \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x}|\mathbf{y})}\right].$$

We can split the ELBO into two terms: the first one represents the expected log-likelihood with respect to the variational posterior, and the second one is the KL divergence between

the variational posterior and the prior, i.e.,

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x}|\mathbf{y})}\right], \\ &= \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log p(\mathbf{y}|\mathbf{x})\right] - \left[\frac{q(\mathbf{y}|\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right], \\ &= \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log p(\mathbf{y}|\mathbf{x})\right] - \mathrm{KL}\left[q(\mathbf{x}|\mathbf{y})\|p(\mathbf{x})\right]. \end{aligned} \tag{5.6}$$

The variational auto-encoder of Kingma and Welling (Kingma & Welling, 2013) is an unsupervised method which places another neural network in the likelihood to model the forward direction. The big difference is that we place a radiative transfer model in the likelihood, fixing a low value of noise variance, thus forcing the forward mapping to follow the physical principles encoded in the RTM. A very similar approach was first reported in (McCarthy et al., 2017). The first term of the ELBO is not analytically tractable since taking the expected value of the likelihood involves integrating over the highly complex RTM. In stead, we perform a Monte Carlo estimation of the expected value (i.e, the first term) (Robert & Casella, 2013). The second term does have a simple analytical form as it is the KL divergence between two Gaussians.

It can easily be shown that maximizing the ELBO corresponds to minimizing the Kullback-Leibler divergence between the variational and the true posterior:

$$\begin{aligned}\mathrm{KL}\left[q(\mathbf{x}|\mathbf{y})\|p(\mathbf{x}|\mathbf{y})\right] &= -\mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x}|\mathbf{y})}\right] \\ &= -\mathbb{E}_{q(\mathbf{x}|\mathbf{y})}\left[\log \frac{p(\mathbf{x},\mathbf{y})}{q(\mathbf{x}|\mathbf{y})} - \log p(\mathbf{y})\right] \\ &= -\mathcal{L} + \log p(\mathbf{y}). \end{aligned} \tag{5.7}$$

Therefore, maximizing $\mathcal{L}$ with respect to ϕ should make $\mathrm{KL}\left[q(\mathbf{x}|\mathbf{y})\|p(\mathbf{x}|\mathbf{y})\right]$ fairly small and hence $\mathcal{L} \approx \log p(\mathbf{y})$. The maximization of $\mathcal{L}$ with respect to the prior parameters, $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{S}\}$, is hence expected to maximize $\log p(\mathbf{y})$, which is the maximum likelihood principle for parameter estimation. In practice, we maximize $\mathcal{L}$ simultaneously with respect to $\boldsymbol{\theta}$ and ϕ.

The previous approach can be easily extended to the case of having several observed data instances $\{\mathbf{y}_i\}_{i=1}^n$. In that case the objective is simply the sum of $\mathcal{L}_i$, for $i = 1, \ldots, n$, where $\mathcal{L}_i$ is the lower bound corresponding to $\mathbf{y}_i$, *i.e.*, the i-th data instance. This sum can be approximated using mini-batches and optimized using stochastic optimization techniques such as the ADAM algorithm (Kingma & Ba, 2015). In the experiments presented here, we use a mini-batch size of 1 for all experiments. Since we use the actual PROSAIL RTM, and not a differentiable emulator, we have to use finite differences in order to compute the gradient of the RTM. For a proof of convergence of stochastic optimization see (Robbins & Monro, 1951). The variational approach is expected to find reasonable values for the prior parameters $\boldsymbol{\theta}$, using approximate maximum likelihood estimation, and to provide a recognition model $q(\mathbf{x}|\mathbf{y})$ that can be used to infer the potential values of $\mathbf{x}$ given $\mathbf{y}$.

5.3 Monte Carlo expectation maximization

Another method which can be used to address the learning goals described in Section 5.1, i.e. to infer the prior parameters from the observed data, and to generate samples from the posterior distribution $p(\mathbf{x}|\mathbf{y})$, is the Monte Carlo expectation maximization (MCEM) method (Wei & Tanner, 1990).

We begin by briefly describing the expectation maximization (EM) algorithm, which can be used to maximize the likelihood function in models that involve latent variables (Dempster et al., 1977). This is precisely the scenario considered in Section 5.1. Namely, given some observed data $\{\mathbf{y}_i\}_{i=1}^n$, we would like to maximize

$$\prod_{i=1}^{n} p(\mathbf{y}_i|\boldsymbol{\theta}) = \prod_{i=1}^{n} \int_{\mathcal{C}} p(\mathbf{y}_i|\mathbf{x}_i) p(\mathbf{x}_i|\boldsymbol{\theta}) d\mathbf{x}_i \,, \tag{5.8}$$

as a function of the prior parameters $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{S}\}$. Direct optimization of (5.8) is intractable, since we cannot marginalize the latent variables $\mathbf{x}_i$. The EM algorithm uses the fact that the complete likelihood function $p(\mathbf{y}_i, \mathbf{x}_i|\boldsymbol{\theta}) = p(\mathbf{y}_i|\mathbf{x}_i) p(\mathbf{x}_i|\boldsymbol{\theta})$ is tractable. Consider the following decomposition of the logarithm of (5.8)

$$\sum_{i=1}^{n} \log p(\mathbf{y}_i|\boldsymbol{\theta}) = \sum_{i=1}^{n} \mathcal{L}(q_i, \boldsymbol{\theta}) + \mathrm{KL}(q_i \| p_i) \,, \tag{5.9}$$

where we have introduced an approximate distribution $q_i(\mathbf{x}_i)$ and

$$\mathcal{L}(q_i, \boldsymbol{\theta}) = \int_{\mathcal{C}} q_i(\mathbf{x}_i) \log \frac{p(\mathbf{y}_i, \mathbf{x}_i|\boldsymbol{\theta})}{q_i(\mathbf{x}_i)} d\mathbf{x}_i \,, \tag{5.10}$$

$$\mathrm{KL}(q_i \| p_i) = -\int_{\mathcal{C}} q_i(\mathbf{x}_i) \log \frac{p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta})}{q_i(\mathbf{x}_i)} d\mathbf{x}_i \,. \tag{5.11}$$

This holds because

$$\begin{aligned} \mathcal{L}(q_i, \boldsymbol{\theta}) + \mathrm{KL}(q_i \| p_i) &= \int_{\mathcal{C}} q_i(\mathbf{x}_i) \log \frac{p(\mathbf{y}_i, \mathbf{x}_i|\boldsymbol{\theta})}{p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta})} d\mathbf{x}_i \\ &= \int_{\mathcal{C}} q_i(\mathbf{x}_i) \log p(\mathbf{y}_i|\boldsymbol{\theta}) d\mathbf{x}_i = \log p(\mathbf{y}_i|\boldsymbol{\theta}). \end{aligned}$$

Note that (5.11) is the Kullback Leibler divergence between q_i and the exact posterior $p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta})$ for the instance $\mathbf{y}_i$.

The EM algorithm maximizes (5.9) in a two stage iterative process. Assume the current parameter vector is $\boldsymbol{\theta}^{\mathrm{old}}$. In the E step, the lower bound $\sum_{i=1}^n \mathcal{L}(q_i, \boldsymbol{\theta}^{\mathrm{old}})$ is maximized with respect to each q_i, assuming $\boldsymbol{\theta}^{\mathrm{old}}$ to be fixed. Because $\sum_{i=1}^n \log p(\mathbf{y}_i|\boldsymbol{\theta})$ does not depend on each q_i, the solution to this problem consists in setting each $q_i(\mathbf{x}_i)$ equal to $p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta}^{\mathrm{old}})$, minimizing $\mathrm{KL}(q_i \| p_i)$ in consequence. In the subsequent M step, each $q_i(\mathbf{x}_i)$ is held fixed, and $\sum_{i=1}^n \mathcal{L}(q_i, \boldsymbol{\theta}^{\mathrm{old}})$ is maximized with respect to $\boldsymbol{\theta}$, to give new prior parameters $\boldsymbol{\theta}^{\mathrm{new}}$. This will cause the lower bound $\sum_{i=1}^n \mathcal{L}(q_i, \boldsymbol{\theta}^{\mathrm{old}})$ to increase, which will in turn increase the log-likelihood $\sum_{i=1}^n \log p(\mathbf{y}_i|\boldsymbol{\theta})$. Critically, $q_i(\mathbf{x}_i)$ will be computed in this step using

$\boldsymbol{\theta}^{\text{old}}$, which is fixed. Therefore, the only required integral to evaluate in the M step is

$$\mathcal{L}(q_i, \boldsymbol{\theta}) = \int_{\mathcal{C}} q_i(\mathbf{x}_i) \log p(\mathbf{x}_i|\boldsymbol{\theta}) d\mathbf{x}_i + \text{const.} \tag{5.12}$$

A difficulty, however, is that the posterior $p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta}^{\text{old}})$ is intractable, which makes computing q_i and hence the integral in (5.12) challenging. Monte Carlo EM (MCEM), provides a solution to this problem (Wei & Tanner, 1990). The intractable integral in (5.12) is simply approximated by a Monte Carlo average over several samples drawn from q_i. Namely,

$$\mathcal{L}(q_i, \boldsymbol{\theta}) \approx \frac{1}{S} \sum_{s=1}^{S} \log p(\mathbf{x}_i^s|\boldsymbol{\theta}) + \text{const.}, \tag{5.13}$$

where $\mathbf{x}_i^s$ has been generated from q_i and S is the number of generated samples. The convergence properties of MCEM are analyzed in (Neath, 2013).

Recall that the approximate distribution q_i is targeting the exact posterior $p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta}^{\text{old}})$. So ideally, we should generate the samples $\mathbf{x}_i^s$ from the exact posterior. For this, we use Hamilton Monte Carlo (HMC) (Neal et al., 2011) as in (Kingma & Welling, 2013). HMC is a Markov chain Monte Carlo (MCMC) method that can be used to generate (correlated) samples from some target distribution (Martino & Elvira, 2017). More specifically, a Markov chain is generated whose stationary distribution coincides with the target distribution. By running the Markov chain for a sufficiently large number of steps one can obtain an approximate independent sample from $p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta}^{\text{old}})$. HMC has the advantage that, when well-tuned, it substantially reduces the correlation among samples (Martino & Elvira, 2017). In order to do this, it simulates a dynamical system that uses information about the gradient of the posterior, *i.e.*, $\nabla_{\mathbf{x}_i} \log p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta}^{\text{old}})$, to sample from regions of high posterior probability. In our implementation of MCEM, the HMC procedure consists of 20 leapfrog steps with small step-size (*i.e.*, 5×10^{-4}) which guarantees a sufficiently high acceptance rate. In practice, we only use just one sample to approximate (5.13). Each time, the Markov chain is initialized at the mode of the posterior distribution, which is found using quasi-newton optimization methods (*i.e.*, L-BFGS). Of course, after optimizing the prior parameters $\boldsymbol{\theta}$ using MCEM, HMC can be used to generate samples from the approximate posterior distribution $\widehat{p}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) \propto p(\mathbf{y}|\mathbf{x})\widehat{p}_{\boldsymbol{\theta}}(\mathbf{x})$.

5.4 Considerations for method choice

Note that both, the variational and MCEM methods, provide an estimation of the parameters $\boldsymbol{\theta}$ of the prior. Thus, we obtain a Gaussian approximation of the prior, which is denoted here as $\widehat{p}_{\boldsymbol{\theta}}(\mathbf{x})$. Therefore, both techniques provide the following posterior approximation

$$\widehat{p}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) \propto p(\mathbf{y}|\mathbf{x})\widehat{p}_{\boldsymbol{\theta}}(\mathbf{x}). \tag{5.14}$$

The variational algorithm, however, provides another posterior approximation given in Eq.(5.5), i.e.,

$$q(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\text{NN}}(\mathbf{y}), \boldsymbol{\Sigma}_{\text{NN}}(\mathbf{y})), \tag{5.15}$$

which yields an important advantage with respect the previous one: given an observation $\mathbf{y}$, using $q(\mathbf{x}|\mathbf{y})$ we can easily, and at low computational cost, produce a predictive mean $\boldsymbol{\mu}_{\text{NN}}(\mathbf{y})$ and covariance $\boldsymbol{\Sigma}_{\text{NN}}(\mathbf{y})$. The approximation $\widehat{p}(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ on the other hand would require the use of additional Monte Carlo schemes for obtaining a predictive mean and variance, for each new observation vector $\mathbf{y}$. Another advantage of the variational approach is the computational cost of training compared to the MCEM method. The MCEM scheme, however, is able to handle more practical scenarios (*e.g.*, problems involving multiple posterior modes, heavy tailed distributions, etc.), leading to better performance in terms of smaller error in the parameter estimation of the prior. The variational approach described here would require a different and more general derivation for addressing these scenarios, see *e.g.* (Mescheder et al., 2017). These features of each method are confirmed by the results obtained in our experiments.

5.5 Experiments

We illustrate the strengths and weaknesses of the two approaches, first by means of informative toy experiments: One that studies the computational efficiency of the respective methods, and another which analyzes their ability to handle forward models leading to *multimodal posteriors*. Following this, we show how these approaches can be used to perform inference over biophysical parameters using an RTM as the forward model.

5.5.1 Marginal likelihood estimation by reverse importance sampling

In order to evaluate the performance of the methods described in the sections below, we use an estimator of marginal likelihood on a test-dataset. More precisely, we use the Reverse Importance Sampling (RIS) estimator (Llorente et al., 2020) described below:

1. Sample L values $\{\mathbf{x}_l\}_{l=1}^{L}$ from the posterior with an MCMC-method. We use Hamiltonian Monte Carlo.
2. Fit at density estimator $q(\mathbf{x})$ to the samples $\{\mathbf{x}_l\}_{l=1}^{L}$. In this work we fit a Gaussian mixture model, doing cross validation in order to find the best number of components.
3. Sample M new values $\{\mathbf{x}_m\}_{m=1}^{M}$ from the posterior to be inserted in the following estimator:

$$p(\mathbf{y}) \simeq \left(\frac{1}{M} \sum_{m=1}^{M} \frac{q(\mathbf{x}_m)}{p(\mathbf{x}_m)p(\mathbf{y}|\mathbf{x}_m)} \right)^{-1} \quad \text{where} \quad \mathbf{x}_m \sim p(\mathbf{x}|\mathbf{y}).$$

For the proof and more details, see (Llorente et al., 2020). It is important to remark that the function $q(\mathbf{x})$ must be a valid probability density for which there are several possible choices. If one chooses $q(\mathbf{x}) = p(\mathbf{x})$, RIS becomes the so-called harmonic mean estimator (this name is due to the fact that the corresponding estimator is the harmonic mean of the likelihood values). However, it has been shown that this does not lead to a good estimator. It is possible to show that, in order to ensure finite variance of the resulting estimator, the density $q(\mathbf{x})$ should have equal or lighter tails than the posterior $p(\mathbf{x}|\mathbf{y})$ (e.g., see first numerical example in (Llorente et al., 2020)). Gaussian mixture approximations and kernel density estimators of $p(\mathbf{x}|\mathbf{y})$ are suitable choices for $q(\mathbf{x})$. Different alternative estimators of the marginal likelihood are possible mixing MCMC and importance sampling schemes (see (Llorente et al., 2020; Martino et al., 2017)).

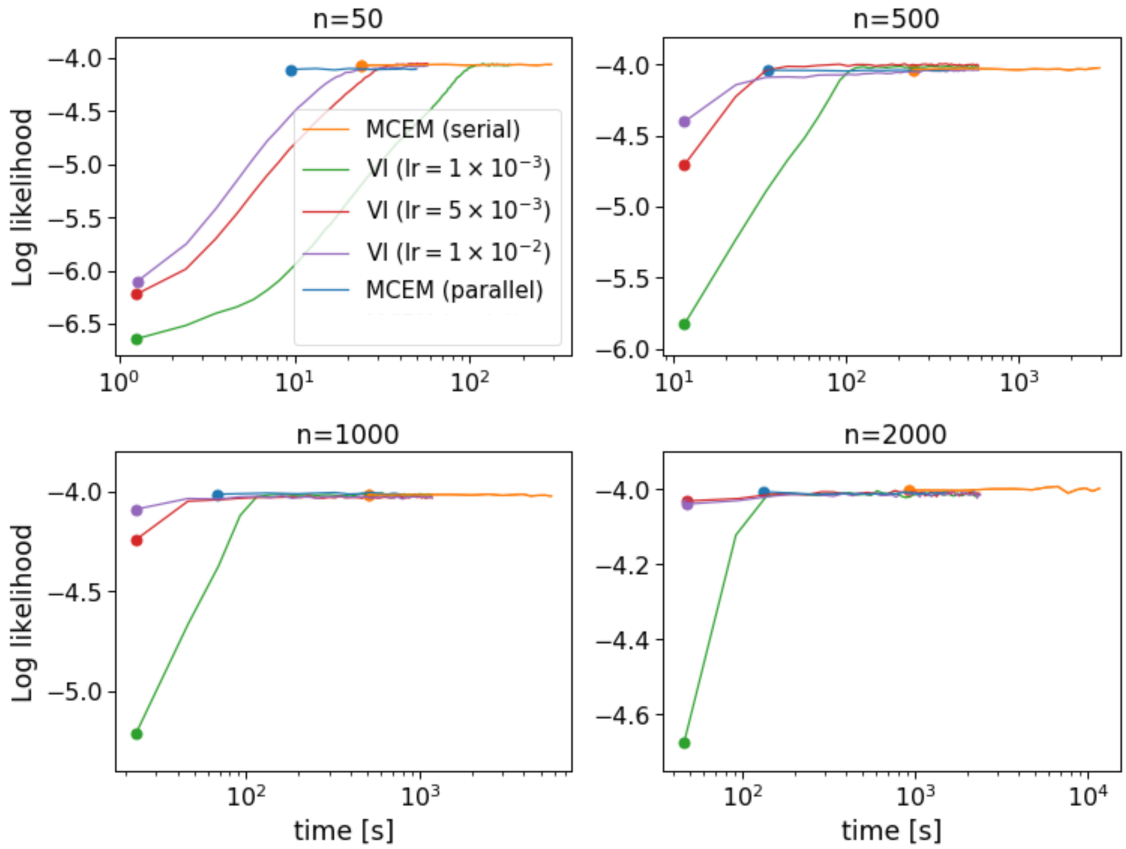


Figure 5.1: Marginal log-likelihood of test dataset as a function of training time for different inference methods. The MCEM algorithm may be parallelised which speeds up computational speed considerably. Four different sizes of training datasets are used, showing that the VI method is computationally more efficient than the MCEM method for larger datasets.

5.5.2 On the computational efficiency

In order to analyze the computational efficiency of the two approaches, we consider a simple forward model (with $\mathbf{x} = [x_1, x_2]$)

$$\mathbf{f}(\mathbf{x}) = f(x_1, x_2) = [2x_1, 2x_2],$$

for which both approaches converge to the true values of the parameters of the prior. We draw the training data $\{\mathbf{x}_i\}_{i=1}^n$ from the prior

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix}4\\6\end{bmatrix}, \begin{bmatrix}1 & 0.6\\0.6 & 1\end{bmatrix}\right),$$

and pass it through the mapping $\mathbf{f}$ in order to generate the training data $\{\mathbf{y}_i\}_{i=1}^n$. Datasets of several sizes $n = \{50, 500, 1\,000, 2\,000\}$ are used for training the models. The model likelihood noise is in all experiments fixed at a negligible value, with $\sigma^2 = 10^{-7}$, in order to reflect the trust in the knowledge encoded in the RTMs.

In Fig. 5.1 we plot an estimate of the average log marginal likelihood of each method

on a test dataset as a function of training time (averaged over 40 repetitions). The marginal likelihood is computed using the estimator described in Sec. 5.5.1. Observing the test log-likelihood, which is computed after each epoch, we see that the MCEM method convergences after 1 epoch (one iteration of the E and M steps). With a training dataset of 50 points, each epoch of training is sufficiently fast that a parallelized version of the MCEM method (in which each E step is done in parallel) converges faster than the VI method. For the non-parallelized algorithm, this is not the case. For larger datasets, however, VI converges before the completion of 1 epoch of the MCEM algorithm. Since this is a simple toy problem, larger learning rates can be used in the VI method, leading to earlier convergence (just after 1 epoch) for datasets of 1 000 and 2 000 points. We can conclude from these experiments that the VI approach (as a consequence of stochastic optimization) has a better scaling properties with respect to the dataset size than MCEM.

5.5.3 Dealing with multimodal posteriors

We have seen that when faced with sufficiently large amounts of training data, variational inference performs faster than Monte Carlo sampling methods. However, since the form of the variational posterior assumed in Eq. (5.5) is unimodal, we cannot expect it to be able to capture any multimodality in the true posterior. Consider for instance the forward mapping

$$\mathbf{f}(\mathbf{x}) = [x_1^2, x_1 x_2].$$

For a given observed $\mathbf{y} = [y_1, y_2] \geq [0, 0]$ there will always be two possible solutions, namely $\mathbf{x}^{(1)} = [\sqrt{y_1}, y_2/\sqrt{y_1}\ ]$ and $\mathbf{x}^{(2)} = [-\sqrt{y_1}, -y_2/\sqrt{y_1}\ ]$, making the posterior inherently multimodal. As stated in the previous sections, we consider a low noise value $\mathbf{e} \sim \mathcal{N}(\mathbf{e}|\mathbf{0}, 10^{-7}\mathbf{I})$. In this example, the prior density is Gaussian with parameters

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix} \right),$$

from which 500 samples are drawn and passed through $\mathbf{f}$ to generate the training dataset.

In the process of maximizing the ELBO, the expected log-likelihood with respect to the variational posterior is computed. We can see from Fig. 5.2, however, that the variational posterior, upon convergence, only captures the positive mode at $\mathbf{x}' = [2, 2]^\top$ of the true posterior given the observation $\mathbf{y}' = [4, 4]^\top$. On the other hand, the MCEM algorithm computes the expected complete log-likelihood with respect to the true posterior as approximated with HMC. As opposed to the variational posterior, HMC does manage to capture both the modes of the true posterior as shown in Fig. 5.2. The learning algorithm of the MCEM method is therefore more likely to converge to the true parameters of the prior if the posterior is multimodal.

We can see the inability of the variational method to capture the multimodality of the problem from the results of the converged methods given in Table 5.1. The fitted parameters of the prior are far from the true ones when compared to the results of the MCEM method which is also reflected in the KL divergence between the fitted and true prior distributions. Multimodality such as this is likely to exist in the remote sensing experiment below, as it has been remarked before that different configurations of inputs can lead to the same output making it an *ill-posed inversion problem* (Gómez-Dans et al.,

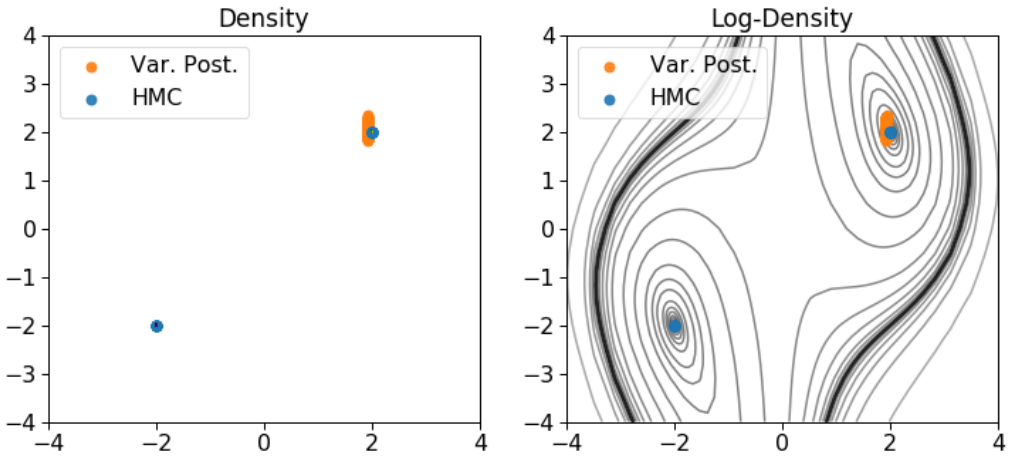


Figure 5.2: Contour plots of samples from the true posterior conditioned on the observation $\mathbf{y} = [4, 4]^\top$. HMC samples using the prior parameters learned by the MCEM method shown in blue, and samples from the learned variational posterior in orange. The density (left) is so sharply peaked around the two modes that it is more informative to study the log-density (right).

2016).

Method	**VI**	**MCEM**
Mean	$\begin{bmatrix} 1.20 \\ 1.76 \end{bmatrix}$	$\begin{bmatrix} 1.027 \\ 2.061 \end{bmatrix}$
Covariance	$\begin{bmatrix} 0.619 & 0.682 \\ 0.682 & 1.600 \end{bmatrix}$	$\begin{bmatrix} 1.001 & 0.617 \\ 0.617 & 0.945 \end{bmatrix}$
D_{KL}	0.315	0.00581

Table 5.1: Comparison of methods for inference on a forward model which leads to a bimodal posterior. The first and second rows show the estimated mean vector and covariance matrix of the prior. The third row shows the KL divergence between the fitted and the true prior.

5.5.4 PROSAIL experiment

We now turn to inference in a remote sensing setting using one of the most widely used RTMs over the last two decades in the field (Jacquemoud et al., 2009) as our physical forward model. We consider PROSAIL for simulating Landsat-8 spectra. Landsat-8's Operational Land Imager (OLI) includes nine spectral bands with wavelengths ranging from 0.433 μm to 1.390 μm, leaving us with an output-dimension of $P = 9$ for our problem. In our experimental setup, we have chosen to work with the most relevant leaf-level parameters to monitor vegetation status and functioning included in PROSAIL, namely water content (Cw), dry matter content (Cm) and chlorophyll content (Chl), resulting in an input dimension of $D = 3$. The remaining parameters were set constant during our experiments and their values were obtained from previous studies to be representative of

realistic cases. For details see (Svendsen et al., 2020a).

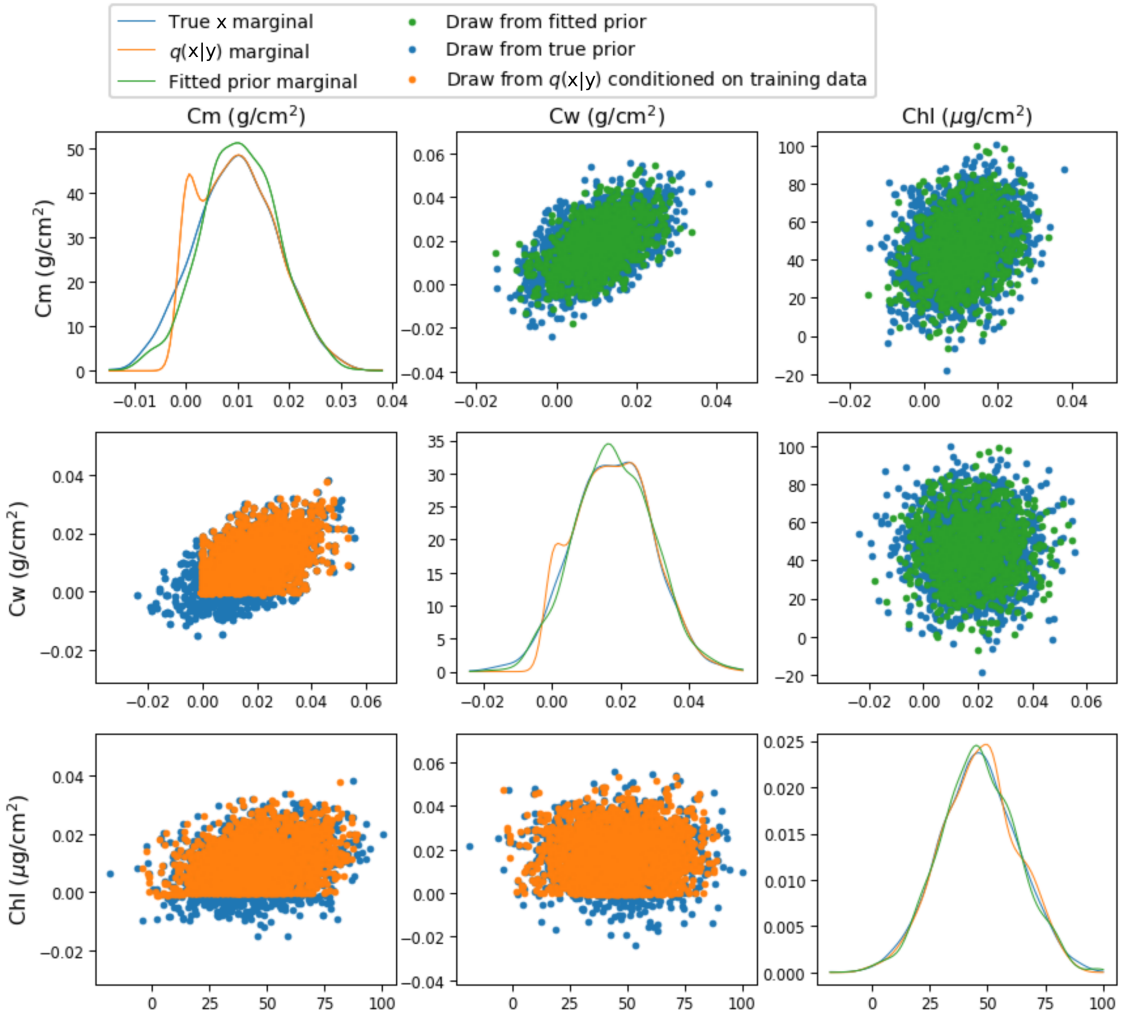


Figure 5.3: Results of the variational approach to inference over PROSAIL. The blue points are **x**'s from the training set, while the green points are draws from the fitted prior. The orange points are draws from the variational posterior conditioned on the training **y**'s. The diagonal shows KDE plots of **x** using samples from the ground truth prior (blue), the variatonal posterior conditioned on training data (orange) and the fitted prior (green).

Constraining the radiative transfer models with realistic and representative distributions of their inputs is a key part of the RTM inversion process. To facilitate this, in this work we relied on the largest global plant traits database available, the TRY database (Kattge et al., 2011; Kattge et al.), which contains thousands of leaf data records measured at unprecedented spatial and climatological coverage. Using these data we computed the following empirical mean vector and covariance matrix which was then used to sample 2 000 values of **x** and pass them through PROSAIL to generate the training data. The empirical mean and covariance (to be compared with the results in Table 5.5.4) of the

samples are

$$\hat{\mathbf{m}} = \begin{bmatrix} 9.76e\text{-}3 \\ 1.77e\text{-}2 \\ 46.2 \end{bmatrix}, \hat{\mathbf{S}} = \begin{bmatrix} 6.42e\text{-}5 & 5.06e\text{-}5 & 3.68e\text{-}2 \\ 5.06e\text{-}5 & 1.34e\text{-}4 & -2.86e\text{-}3 \\ 3.68e\text{-}2 & -2.86e\text{-}3 & 288 \end{bmatrix}.$$

The units of the parameters are g/cm^2 for Cm and Cw, and μg/cm^2 for Chl respectively. Note that the ground truth prior estimated from the TRY database has some probability density in the negative region of parameter space. This is not physically meaningful, but serves the point of illustrating the capabilities of the inference methods. We alter PROSAIL so that it sets every negative parameter to 0 before mapping into spectral space which actually implies a modified likelihood that will lead to more multimodality (since all negative values in $\mathbf{x}$ will be mapped into the same value, i.e. 0, and then through PROSAIL into a spectrum).

Method	**VI**	**MCEM**
Mean	$\begin{bmatrix} 1.02e\text{-}2 \\ 1.80e\text{-}2 \\ 45.9 \end{bmatrix}$	$\begin{bmatrix} 1.01e\text{-}2 \\ 1.81e\text{-}2 \\ 46.6 \end{bmatrix}$
Covariance	$\begin{bmatrix} 5.32e\text{-}5 & 4.31e\text{-}5 & 3.52e\text{-}2 \\ 4.31e\text{-}5 & 1.19e\text{-}4 & 2.18e\text{-}3 \\ 3.52e\text{-}2 & 2.18e\text{-}3 & 280 \end{bmatrix}$	$\begin{bmatrix} 5.49e\text{-}5 & 4.45e\text{-}5 & 3.31e\text{-}2 \\ 4.45e\text{-}5 & 1.25e\text{-}4 & -1.74e\text{-}3 \\ 3.31e\text{-}2 & -1.74e\text{-}3 & 292 \end{bmatrix}$
D_{KL}	0.0208	0.0123

Table 5.2: Comparison of methods for inference on biophysical parameters using a radiative transfer forward model. The first and second columns show the mean vector and covariance matrix respectively of the true and the estimated prior over the causes, using E notation for space. The third column shows the Kullback-Leibler divergence between the fitted and the true prior.

The results of the variational approach to inference over PROSAIL are summarized in Fig. 5.3. We see that the parameters of the prior are fitted well, which can also be confirmed in Table 5.5.4 quantitatively, even though the variational posterior is not able to produce predictive means in the negative domain. It is interesting to note that the modification of PROSAIL to truncate negative data, which leads to multimodality, does not prevent the variational approach from estimating the parameters of the prior well.

Nevertheless, the MCEM method is somewhat more accurate than the VI method, obtaining a KL divergence with to the true prior of 1.23×10^{-2} compared to 2.08×10^{-2} obtained using the VI approach. This is to be expected since, as we have seen, the MCEM approach handles multimodality better. We especially foresee a clear difference in results in future work the LAI variable which is difficult to estimate due to its multimodal posterior distribution as has been pointed out in the literature (Gómez-Dans et al., 2016).

Once the VI method has converged, the neural network which parameterizes the variational posterior can be used as a fast inverse model that maps from observed satellite

spectra to biophysical variables. Using the mean outputs that model the mean value of the variational posterior we can obtain good predictive accuracy on a test set as shown in the scatter plots of Fig. 5.4. Despite the promising results, it is very important to note that we run our experiments using a simplified PROSAIL configuration, keeping some of the input parameters static, and that results can vary greatly in more realistic modelling scenarios.

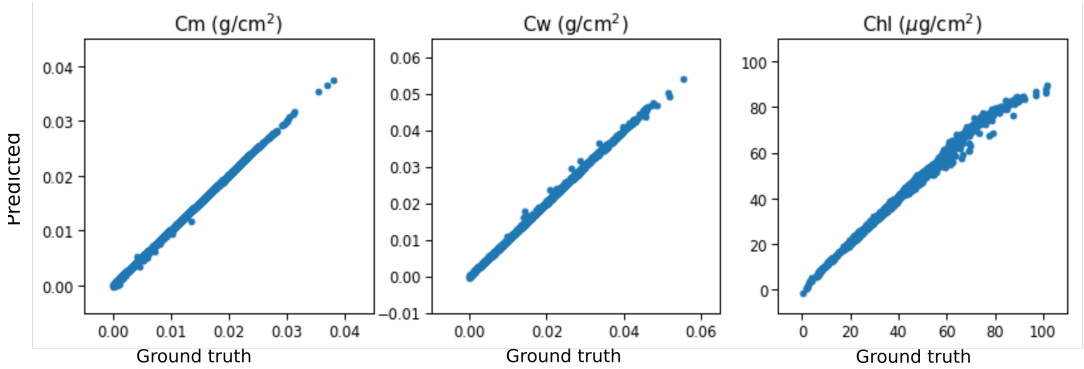


Figure 5.4: True values of RTM parameters in test dataset versus mean of variational posterior conditioned on spectra in test dataset. The trained encoder network can thus be used as an effective predictive model

5.6 Concluding remarks

In this chapter we have approached the inverse problem of estimating biophysical parameters from observational reflectances, focusing not only on parameter point estimates but on their full multivariate distribution. Probabilistic inverse modelling, although not so widely used in remote sensing applications, has proven to be a powerful tool, providing more general (and hence potentially more valuable) solutions than point-wise approaches, and can help in better understanding the problem itself (Zhang et al., 2005; Coccia et al., 2015; Ma et al., 2017). We evaluated two different approximations that include an RTM forward model to enforce the inverse estimations to be physically consistent.

Both of the applied techniques have different advantages and shortcomings which we illustrated with toy examples and with simulations from the PROSAIL RTM. The MCEM-based approach admits more flexible models while the VI method is computationally more efficient. For instance, while MCEM deals easily with multimodal distributions, this is a challenge for VI. On the other hand, the convergence time of the VI approach is several orders of magnitude faster, depending on the problem. Moreover, the VI scheme provides a posterior approximation, with a predictive mean and a covariance matrix, implicitly defined by the trained neural network that can be readily evaluated. The experiment involving PROSAIL shows that, while the accuracy of the VI and MCEM methods are deemed similar, the computational simplicity of the VI approach is critical in this problem. Note that including the PROSAIL RTM in the forward-inverse modelling loop increases the time of computation, and combining it with MCEM makes it unfeasible, especially for large data sets. This can be helped by using an emulator as described in the previous chapter. These are not only faster to evaluate but differentiable, removing the need for using finite differences.

6. Conclusion and Discussion

This thesis has presented different ways of applying probabilistic models in conjunction with physics knowledge in order to develop machine learning algorithms tailored to specific remote sensing problems. One of the main tendencies in the machine learning literature is to develop model-agnostic algorithms that learn patterns purely from data. Although such algorithms are trained on large datasets, they can still produce predictions which are inconsistent with the laws of physics. On the other hand, there is a growing interest in the intersection between the fields of physics and machine learning (Willard et al., 2020) and how methods from one can be used to improve methods from the other. Among the different directions of research within this intersection, this thesis has focused on improving inverse modelling by encoding physical knowledge in machine learning regression, and on improved emulation of simulators.

On inverse modelling with Gaussian process models

We started off by showing how the inverse problem of parameter retrieval from satellite-observed reflectances, which is often approached using GP regression (Camps-Valls et al., 2016), can be tackled more efficiently using deep GP regression. GPs are frequently used in remote sensing for a variety of reasons. They constitute a probabilistic treatment of regression problems leading to an analytical expression for the predictive uncertainty which is an attractive feature (Verrelst et al., 2013; Schneider et al., 2014). This also allows for effective error propagation from the inputs to the outputs as has recently been shown in (Johnson et al., 2019). Furthermore, GPs are not pure black box models because, through the use and design of appropriate covariance functions, one can include prior knowledge about the signal characteristics (e.g. nonstationarity, heteroscedasticity, etc.).

Gaussian process regression does, however, have two big drawbacks in their poor scaling to large datasets and the inability to model complex hierarchical structure using standard kernels (such as those described in Sec. 2). The problem of computational cost is ameliorated through the use of sparse GPs which enjoy training costs that scale linearly with the number of data-points. The doubly stochastic variational inference deep GP algorithm applied in this work also boasts a linear training cost and is able to model more

complex data. For the problem of biophysical parameter retrieval from infrared sounder data, we showed that the DGP model outperforms sparse and full GP models in terms of predictive accuracy and quality of uncertainty estimates. The full paper (Svendsen et al., 2020b), which may be found in the Appendix, also shows how DGPs favorably compare with the state-of-the-art neural network method of (Hieronymi et al., 2017) for ocean color parameter retrieval using Sentinel-3 satellite data.

As hybrid systems are developed where the output of machine learning models are passed to physical models, improved uncertainty estimation become more important in order to perform correct error propagation. Furthermore, the increasing stream of available remote sensing data calls for scalable machine learning algorithms. For these reasons we foresee an increased use of DGP regression in the field of biophysical parameter retrieval in the future.

On encoding physics knowledge in Gaussian process regression

We first developed a GP framework for jointly modelling in-situ and simulated data without impairing the predictive performance on the in-situ data. This was accomplished through the introduction of a trust parameter which modelled the relative noise-variance between the data-sources. Fitting the hyperparameters by maximizing the leave-one-out likelihood over the in-situ data in particular ensured that the method did not overfit to the simulated data. The resulting scheme was both simple and robust, and improved the predictive performance on the in-situ data, especially in scenarios of extrapolation. The framework extends to arbitrarily many datasources and can be used on data generated with different sensors or with different simulators.

We then studied the latent force models of (Alvarez et al., 2009) for modelling the dynamics of remotely sensed soil moisture data. This was a conceptually different way of encoding expert knowledge as it builds the GP model around a set of assumptions about the physical system at hand. More precisely, that the outputs are governed by a first order ODE, the forcings of which are Gaussian processes. From this assumption a multi-output kernel is derived with physically meaningful hyperparameters. We showed how the latent forces that are fitted during the training phase match well with the independently measured precipitation timeseries. This is a promising result as the ability to monitor rainfall from satellite measurements would help in the understanding of the hydrological cycle of water passing through ocean, land, and atmosphere. Furthermore, applications within any physical system that can be measured with remote sensing can benefit from this type of modelling if the dynamics are sufficiently well described by an ordinary differential equation.

On active learning for emulation of RTMs

Simulators of physical processes are usually computationally expensive, often to the extent that it makes applications such as sensitivity analysis impossible. Furthermore, simulator codes are typically full of heuristics and logic statements that make analysis of derivatives intractable. This makes it hard to use such models in loss functions that one tries to optimize, e.g. the one seen in Sec. 5.2. Emulators provide a solution to these problems as predictions of machine learning methods often consist of fast operations such as matrix multiplications and are differentiable, depending on the model.

We presented an active learning approach which included information about the gradient of the underlying function in the acquisition function. This targets not only regions with

few data-points, but also regions of high function variability. We showed that our method builds a more accurate emulator of a simplified PROSAIL RTM using fewer runs of the simulator than when using random sampling from a physically meaningful prior. We also compared our sampling scheme to Sobol sequence and Latin hypercube sampling (currently used in many applications), showing improved performance.

Among the existing radiative transfer models, PROSAIL is not the most computationally expensive. Our method is especially useful when each run of the simulator takes a considerable amount of time, as is the case for the MODTRAN RTM (Berk et al., 1987). In such cases it becomes crucial to use information learned about the RTM in order to evaluate it a minimum number of times.

On performing approximate inference over RTMs

In the final chapter we took an approximate Bayesian approach to the inversion problem. Approximate in the sense that we, along with a likelihood function, define a *family of priors* in stead of a specific prior which is usually the case in Bayesian inference. We fit the parameters of the proposed prior (Gaussian in this case) and thus learn about joint prior distribution over the physical parameters of interest. The likelihood is assumed a Gaussian with low noise, centered around the output of the PROSAIL RTM. This means that the posterior acts as a sort of inverse model, taking observed reflectances and outputting a distribution over physical parameters.

Since the PROSAIL RTM is non-linear, simple marginal likelihood inference is unfeasible. We presented two ways of performing approximate inference in stead, each with their own strengths and weaknesses. The first is based on the variational autoencoder (Kingma & Welling, 2013), where the decoder network is replaced with the PROSAIL RTM. We showed that this variational inference scheme is quicker to train. Furthermore, once trained the variational posterior, parameterized by a neural network, is very fast to evaluate, serving as a swift inverse model. This inference scheme, however, is at a loss when the true posterior is multimodal, as we illustrated in a toy example. This does not end up being a big problem when performing inference over the RTM. The second inference scheme used was that of Monte Carlo expectation maximization. The training phase involves sampling from the true posterior with the Hamilton Monte Carlo method, making the method slower. Evaluating the predictive posterior for new observed data also involves Monte Carlo sampling and is therefore much slower than the VI alternative. The MCEM approach, however, can model multimodality in the true posterior leading to better convergence of the algorithm when this is the case. Therefore, this inference method leads to slightly better estimates of the true prior in the experiment of performing inference over PROSAIL.

Inversion problems are ubiquitous in Earth observation, and there are many different forward models which express domain knowledge of various physical systems. A probabilistic approach to such problems can be extremely useful. It allows us to encode expert knowledge in the likelihood function and to choose meaningful families of priors. As opposed to only providing a point estimate, this type of framework describes the joint distribution over physical parameters. It is important to note that there is work in the literature on speeding up Monte Carlo sampling methods and allowing the VI approach to model multimodal posteriors (Martino & Elvira, 2017; Mescheder et al., 2017).

Future work

In the course of this thesis, a range of new and interesting multi-fidelity machine learning models have been published (Pilania et al., 2017; Perdikaris et al., 2017). This framework presents an ideal way of combining real and simulated data, as done with the presented JGP. This makes it very useful for inversion problems in remote sensing where little in-situ data exists, but where the physical processes in the forward directions are well-described by RTMs. We foresee many applications of this type in the remote sensing literature.

The joint GP and the GP active learning scheme presented here are both examples of methods where other types of GP models could be used. As we have seen, the DGP outperforms the shallow GP on some important parameters, and one could imagine using the relatively simple JGP framework for modelling simulated and in-situ presented here in conjunction with DGPs. With regards to active emulation, we plan to provide a survey describing the different schemes proposed in the vast fields of sequential experimental design and Bayesian optimization.

For the variational scheme presented for inference over RTMs, there are improvements that can be made. A mixture of Gaussians would serve as a much more flexible and realistic prior over the physical parameters compared to the simplified Gaussian model assumed in this work. Furthermore, there are improvements to the VAE approach first presented in (Kingma & Welling, 2013) which allows the handling of multimodal posteriors (Mescheder et al., 2017). The inversion over RTMs can often lead to this kind of multimodality and it is therefore an important line of future work.

While the focus of this thesis has been on problems in the field of remote sensing, there are many applications outside RS where the presented methods would be useful. Most natural sciences use simulator codes in order to analyze various systems of interest. These simulators can be used to specify likelihood models and perform Bayesian inference, as shown in this work, or to generate simulated data to improve the regression on in-situ data. If the codes are computationally expensive, as is often the case, they could benefit from parsimonious emulation schemes as the one presented here. The work presented in this thesis therefore has relevance in many other fields of science.

Related work carried out during this thesis

The published papers that make out the main contribution of this thesis can be found in the Appendix. In addition, the work conducted in this thesis has also been presented on several international conferences and indirectly contributed to other scientific publications:

Journal papers

1. **Svendsen, D.H.**, Morales-Álvarez, P., Ruescas, A.B., Molina, R. and Camps-Valls, G., 2020. Deep Gaussian processes for biogeophysical parameter retrieval and model inversion. *ISPRS Journal of Photogrammetry and Remote Sensing, 166*, pp.68-81.
2. **Svendsen, D.H.**, Martino, L., Campos-Taberner, M., García-Haro, F.J. and Camps-Valls, G., 2017. Joint Gaussian processes for biophysical parameter retrieval. *IEEE Transactions on Geoscience and Remote Sensing, 56(3)*, pp.1718-1727.
3. **Svendsen, D.H.**, Martino, L. and Camps-Valls, G., 2020. Active emulation of computer codes with Gaussian processes–Application to remote sensing. *Pattern Recognition, 100*, p.107103.

Related journal papers

1. Camps-Valls, G., Martino, L., **Svendsen, D.H.**, Campos-Taberner, M., Muñoz-Mari, J., Laparra, V., Luengo, D. and Garcia-Haro, F.J., 2018. Physics-aware Gaussian processes in remote sensing. *Applied Soft Computing, 68*, pp.69-82.
2. Camps-Valls, G., Gómez-Chova, L., Laparra, V., Martino, L., Mateo-Garcia, G., Muñoz-Mari, J., **Svendsen, D.H.** and Verrelst, J., 2020. Statistical biophysical parameter retrieval and emulation with Gaussian processes. In *Data Handling in Science and Technology* (Vol. 32, pp. 333-368). Elsevier.

Conference papers

1. **Svendsen, D.H.**, Martino, L., Campos-Taberner, M. and Camps-Valls, G., 2017, July. Joint Gaussian processes for inverse modelling. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 3980-3983). IEEE.
2. **Svendsen, D.H.**, Morales-Álvarez, P., Molina, R. and Camps-Valls, G., 2018, July. Deep Gaussian Processes for Geophysical Parameter Retrieval. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 6175-6178). IEEE.
3. **Svendsen, D.H.**, Martino, L., Vicent, J. and Camps-Valls, G., 2018, July. Multioutput automatic emulator for radiative transfer models. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 4019-4022). IEEE.
4. Camps-Valls, G., **Svendsen, D.H.**, Martino, L., Muñoz-Mari, J., Laparra, V., Campos-Taberner, M. and Luengo, D., 2017, June. Physics-aware Gaussian processes for Earth observation. In *Scandinavian Conference on Image Analysis* (pp. 205-217). Springer, Cham.
5. Martino, L., **Svendsen, D.H.**, Vicent, J. and Camps-Valls, G., 2020, May. Adaptive Sequential Interpolator Using Active Learning for Efficient Emulation of Complex Systems. In ICASSP 2020-2020 IEEE *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3577-3581). IEEE.

Code repositories

In order to facilitate the use and understanding of the methods presented in this thesis, the implementation of each method is shared in the following code repositories:

Chapter 2: github.com/dhsvendsen/DGP4RS
Chapter 3: github.com/dhsvendsen/JGP, github.com/dhsvendsen/LFM4RS
Chapter 4: github.com/dhsvendsen/AMOGAPE
Chapter 5: github.com/dhsvendsen/RTM_VI_MCEM_INFERENCE

Acknowledgements

The research activities leading to this thesis were supported by the European Research Council (ERC) through the ERC-CoG-2014 SEDAL Project under Grant 647423.

7. Summary in Valencian

7.1 Motivació i objectius

La teledetecció és un ampli camp de la ciència què ens permet monitorar i controlar a escala global els oceans, la terra i l'atmosfera, i obtenir informació clau sobre el clima. Els sistemes òptics emprats a bord de satèl·lit o avió permeten mesurar la radiació electromagnètica reflexada per la superfície de la Terra en una alta resolució espectral, és una rica font d'informació sobre els diversos processos que tenen lloc al nostre planeta i té una llarga llista d'aplicacions. Entre aquestes aplicacions trobem aplicacions en aigües i oceans com ara l'estimació de la salinitat de la superfície del mar (Lagerloef et al., 1995), de la qualitat de l'aigua (Ruescas et al., 2018) i el control del gel marí (Spreen et al., 2008); en el sol i la vegetació com ara la predicció del rendiment del cultiu (Mateo-Sanchis et al., 2019), detecció de sequeres (Kogan, 1995) i l'estimació de la captació de carboni de la vegetació (Alton et al., 2007), així com també el monitoratge dels constituents de l'atmosfera com son la emissió i absorció de CO_2 (Tramontana et al., 2016), detecció de núvols (Mateo-García et al., 2017) i estimació de la concentració de gasos traça com l'ozó (Kondratyev & Varotsos, 2002).

Les dues darreres dècades han experimentat un gran augment en l'aplicació d'algorismes d'aprenentatge automàtic en l'observació de la Terra per fer un ús eficient de l'elevat flux de dades de teledetecció. Els algorismes d'aprenentatge estadístic o automàtic (més conegut com *machine learning*, ML), però, solen ser models agnòstics i massa flexibles i, per tant, acaben per no respectar les lleis fonamentals de la física. D'altra banda, en els darrers anys, hi ha hagut un augment de la investigació orientada a incorporar els coneixements de física en algorismes d'aprenentatge estadístic per obtenir solucions interpretables, consistents, i que tinguin sentit físic.

L'objectiu principal d'aquesta tesi és, en línia amb aquesta recent branca d'investigació, explorar diferents maneres de codificar el coneixement físic per proporcionar mètodes d'aprenentatge automàtic adaptats a problemes específics en teledetecció. En esta Tesi Doctoral, aquest objectiu es persegueix explorant la interacció entre el ML i l'observació de la Terra a través de dues direccions d'investigació. D'una banda, incorporar coneixe-

ment físic en algorismes de regressió per tal de millorar el rendiment, la coherència i la interpretabilitat. D'altra banda, es millorarà la qualitat dels emuladors de models físics complexos i sovint matemàticament intractables.

En este treball mostrem com millorar el rendiment i la interpretabilitat per a la inversió de models físics i estimació de paràmetres biofísics de diferents maneres: 1) combinanant dades observacionals i simulacions de forma sinèrgica en la regressió basada en processos gaussians (GPs), 2) utilitzant nuclis derivats a partir d'equacions diferencials ordinàries (EDOs) específics que apliquen consistència física en GPs, 3) realitzant una inferència aproximada bayesiana sobre models de transferència radiativa que ens permeten recuperar les distribucions de probabilitats sobre els paràmetres físics, i 4) aprendre models d'inversió més ràpids i flexibles amb GPs profunds. A més a més, i per tal de construir emuladors més efectius, incorporem els GPs a un esquema d'aprenentatge actiu que fa ús de la informació de geometria i diversitat apresa de les dades, aconseguint així emuladors més compactes i eficients de simuladors físics.

7.2 Regressió no lineal amb processos gaussians

La relació entre els paràmetres d'estat del sistema que observem (paràmetres biofísics al nostre cas) i els espectres obtinguts per les observacions (espectres adquirits pels sensors satel·itals) s'ha demostrat que és clarament no lineal (Camps-Valls et al., 2011). Per tant, la regressió no lineal és una part clau de la recuperació de paràmetres. Els mètodes d'aquesta tesi es basen en gran mesura en la regressió basada en processos gaussians. Això es deu al fet que son models de regressió probabilística bastant flexibles i que ofereixen maneres intuïtives de codificar coneixements a-priori. Aquesta Tesi proporciona una introducció al mètode de regressió de GPs que ja s'utilitza àmpliament per a la recuperació de paràmetres en teledetecció (Camps-Valls et al., 2016). A més, presentem l'ús d'una extensió profunda (deep) dels processos gaussians (Damianou & Lawrence, 2013) per millorar la recuperació de paràmetres

7.2.1 Regressió del procés gaussià

Un procés gaussià és una *distribució de probabilitat sobre funcions*. Al contrari que una distribució gaussiana sobre una variable aleatòria, que es defineix pels seus valors de mitjana i covariància, un GP es determina per que la mitjana i covariància son funcions. La funció de covariància codifica la similitud entre valors de la funció que estem intentant modelar. Quan es considera un nombre finit de valors de la funció, tindran una distribució gaussiana conjunta determinada per les funcions de mitjana i de covariància.

Si a més escollim un model de probabilitat gaussiana, aleshores és fàcil calcular la probabilitat marginal i optimitzar-la per als paràmetres de les funcions de mitjana i de covariància. Utilitzant la manipulació estàndard de gaussianes (vegeu (Bishop, 2006) Eq. (2.96-97)) també podem obtenir la distribució predictiva d'un nou valor de sortida donat l'entrada. Aquesta serà una altra distribució gaussiana amb una predicció mitjana i la seua variació corresponent. El fet que el model GP no només ofereix prediccions per a un input determinat, sinó que també té una forma natural de valorar la incertesa d'una predicció mitjançant la variància predictiva ha convertit els GPs en mètodes molt populars en la teledetecció (Camps-Valls et al., 2016).

7.2.2 Regressió de processos gaussians profunds

Si bé els GPs s'utilitzen àmpliament en la teledetecció, pateixen dos inconvenients importants. En primer lloc, el cost de l'entrenament escala cúbicament amb el nombre de mostres d'entrenament, cosa que fa que el mètode siga prohibitiu per un conjunt de dades gran o inclús moderat. En segon lloc, quan es tracta d'invertir un model de transferència radiativa (RTM), què per construcció té una estructura complexa i jeràrquica, un GP bàsic (amb una única capa i una covariància fixa) no sempre és capaç de modelar les dades. Els processos gaussians profunds (DGPs), proposats per primera vegada a (Damianou & Lawrence, 2013), han demostrat ser capaços de modelar dades més complexes i que es poden entrenar amb un cost computacional lineal amb el nombre de mostres (Salimbeni & Deisenroth, 2017).

Quan s'aplica la regressió basada en GPs d'una sola capa, la sortida s'utilitza directament per modelar la variable resposta. Aquesta sortida, però, es podria utilitzar per definir la posició d'entrada d'un altre GP. Si es realitza aquesta aplicació, un total de L vegades, es dóna lloc a un *Deep Gaussian Process* de L capes. Per a un GP estàndard, el prior gaussià és combinat amb el model de probabilitat gaussià. Això significa que es pot integrar i calcular la probabilitat marginal i la posterior de forma analítica. Per al model DGP, on els valors latents que s'han d'integrar apareixen com a entrades a la capa següent (és a dir, apareixen dins d'una matriu de covariància complexa), la inferència exacta és intractable. Per aquesta raó, utilitzarem el mètode d'inferència variacional (Salimbeni & Deisenroth, 2017).

La inferència variacional (*variational inference*, VI) és un enfocament àmpliament aplicat en la modelització probabilística quan el càlcul del posterior és intractable. Funciona introduint una família paramètrica de distribucions posteriors candidates dins les quals es busca la distribució òptima. Com que trobar el posterior analític implica un procés complicat d'integració de valors latents, es diu que VI transforma el problema d'integració en un d'optimització. Per obtenir més informació sobre l'esquema VI, vegeu (Svendsen et al., 2020b).

7.2.3 Millora de la recuperació de paràmetres amb DGP

En esta primera part experimental, demostrem el funcionamient dels GPs i els DGPs per a la recuperació de paràmetres atmosfèrics a partir d'informació espectral que recopila el sensor *Infrared Atmospheric Sounding Interferometer (IASI)* abord de la sèrie de la constel·lació de satèl·lits MetOp. En concret ens hem centrat en fer models de predicció de temperatura i humitat relativa. Hem comparat els mètodes de GP més utilitzats:

- GP estàndard: regressió de GP d'una sola capa explicada anteriorment. Té un cost computacional molt elevat amb el nombre de dades.
- GP *sparse*: es tracta d'aproximacions a la regressió estàndard del GP on el cost és línial. Concretament, utilitzem les aproximacions *fully independent training conditional (FITC)* i la *scalable variational gaussian process (SVGP)*.
- Deep GP: el model de GP profund anterior amb un màxim de 4 capes.

Hem comparat les precisions en predicció de diferents conjunts i problemes de teledetecció. La regressió estàndard GP només va ser capaç d'utilitzar 10 000 punts pel seu elevat cost. Quan s'utilitza la mateixa quantitat de dades, els mètodes sparse de GP van produir pitjor resultats que els GP estàndard. Tot i això, quan es va permetre aprofitar més dades, van poder superar-los. Els models Deep GP van superar de forma constant els altres. A més, la

precisió predictiu va augmentar significativament amb la creixent profunditat del DGP fins a afegir la quarta capa. També vam demostrar que la variança predictiva del model DGP estava millor calibrada que la dels altres models.

7.3 Incorporació de coneixements de física en la regressió del GP

Hi ha diferents maneres de millorar els algorismes de regressió mitjançant la incorporació del coneixement físic (Willard et al., 2020). Un dels mètodes és incorporar dades simulades d'un model físic en l'esquema d'aprenentatge de l'algorisme. Això pot conduir a un millor rendiment, però no necessàriament aporta millor comprensió del problema. Un altre enfocament és crear un model de ML basat en el coneixement de les equacions que governen el sistema a estudiar. Aquesta aproximació pot aconseguir una certa adaptació del mètode de regressió a les dades particulars i ens permet aprendre paràmetres físics en el procés d'entrenament del model ML. L'inconvenient és que els supòsits del model poden ser massa forts, fent-lo menys flexible i molt probablement no tan precís.

Mostrem un exemple de cada enfocament i com la incorporació del coneixement físic pot conduir a (i) una millora de la regressió en particular en règims d'extrapolació, i (ii) una millor comprensió del sistema físic subjacent. Desenvolupem un marc de GPs que modelitza conjuntament dades *in-situ* i simulades, i pesa els punts d'entrenament en funció de l'origen i rellevància de les dades. Un segon desenvolupament que fem en aquesta tesi fa servir el models de força latent de (Alvarez et al., 2009) per adaptar la regressió del GP a la dinàmica de les estimacions de la humitat del sòl amb detecció remota. En fer-ho, aprenem un forçament latent que es correspon molt fidelment amb la precipitació (enregistrada independentment) mesurada al lloc rellevant.

7.3.1 Processos gaussians conjunts

Les mesures reals *in situ* de paràmetres biofísics, que es combinen amb les dades de satèl·lit per obtenir els parells de dades que necessitem per formar models d'aprenentatge estadístic, són el resultat de campanyes de camp costoses, en recursos i personal. A causa del cost i la velocitat prohibitius amb què es recopilen aquestes dades, hi ha una escassetat d'aquest tipus de dades. D'altra banda, les simulacions proporcionen una forma ràpida i barata d'obtenir parells d'entrada-sortida. Tanmateix, simplement agrupar dades reals i simulades indiscriminadament pot comportar que el model s'ajuste a les dades simulades al estar lliures de soroll. Això pot comportar un pitjor rendiment sobre les dades reals, ja que normalment hi ha una discrepància en la distribució de les dues fonts de dades. Això també es coneix com a *transferència negativa*.

Per modelar conjuntament les dues fonts de dades, introduïm un hiperparàmetre que expressa el soroll relatiu entre elles. Això permet al GP assignar més o menys pes a un punt de dades en realitzar una regressió basada en la seva font. Per tal d'evitar la transferència negativa, quan s'ajusten als hiperparàmetres només maximitzem la probabilitat de les dades *reals*. Aquest procés es tradueix en un model que només utilitza les dades simulades si es consideren útils per predir les dades reals.

7.3.2 Millora de la regressió i l'extrapolació

En primer lloc, mostrem la eficàcia d'aquests mètodes per a la predicció de l'índex d'àrea foliar *(Leaf Area Index, LAI)* a partir d'espectres de Landsat-8 en camps d'arròs

de l'Albufera de València. En este cas podem utilitzar dades simulades del model de transferència radiativa PROSAIL per millorar la regressió amb el nostre GP compost. Comparem amb mètodes que només utilitzen dades simulades o només les dades reals, o agrupant les dades conjuntament de manera indiscriminada. El nostre mètode millora els resultats de regressió, especialment quan es divideixen les dades de manera que el model de regressió haja de fer l'extrapolació a les regions on no hi ha dades *in situ* d'entrenament.

7.3.3 Inferint les forces latents de la humitat del sòl

En segon lloc presentem un nou model GP *derivat directament de les equacions diferencials que governen* el sistema i per tant les que generen les observacions. Ens plantegem el problema de modelar sèries de temps d'humitat del sòl (*soil moisture, SM*) derivades a partir de sensors passius de microones. La humitat del sòl és una variable d'estat hidrològic clau, important per a la comprensió de diversos processos climatològics i meteorològics (Babaeian et al., 2019).

S'ha demostrat que una ODE de primer ordre és un model útil per a modelar la dinàmica de la humitat del sòl (Delworth & Manabe, 1988), capturant el comportament de descomposició exponencial que presenten les dades SM. Suposem que les observacions es regeixen per una equació diferencial ordinària de primer ordre amb un forçament de GP. En aquest treball emprem models de força latent (*latent force models, LFMs*) (Alvarez et al., 2009) que mostren que la solució d'aquest ODE és en si mateixa un procés gaussià sobre les sortides amb un nucli multi-sortida que conté paràmetres de l'ODO subjacent.

En la part experimental, hem modelat conjuntament la sèrie de temps SM estimada mitjançant tres satèl·lits diferents: Soil Moisture Ocean Salinity (SMOS), Advanced SCATterometer (ASCAT) i Advanced Microanning Scanning Radiometer 2 (AMSR2), i hem demostrar que el marc LFM és molt eficaç per omplir buits, però també que les forces latents derivades tenen una interpretació física diferent. Les forces latents inferides per encaixar el model es correlacionen bé amb les mesures independents de precipitació *in situ*. Això té molt sentit, ja que és, de fet, el contribuent dominant de SM en el sistema.

7.4 Millorar l'emulació de les RTM amb aprenentatge actiu

Molts camps científics utilitzen simulacions de codis informàtics per analitzar sistemes d'interès. Els *simuladors* actuen com a aproximacions convenients a la realitat, permetent-nos estudiar com es distribueixen les malalties entre una població, com es reparteixen les mercaderies en una cadena de distribució, o com interactua la llum amb l'atmosfera de la Terra, per anomenar algunes aplicacions. Hi ha, però, dues limitacions importants associades als simuladors:

- *Cost computacional:* En un intent de captar la veritable mecànica del sistema d'interès, les implementacions numèriques de les equacions que governen el sistema poden arribar a ser complicades, tant de definir com d'implementar computacionalment. Això no és desitjable, ja que dificulta la capacitat de realitzar simulacions exhaustives i anàlisis de sensibilitat (Sobol, 1993).
- *Tractabilitat matemàtica:* els codis informàtics sovint es basen en dècades de desenvolupament iteratiu fent ús de diverses heurístiques que milloren la precisió però fan que els models siguin menys matemàticament tractables i transparents. És especialment útil poder accedir a les derivades i jacobians del model perquè això permet

estudiar la propagació de la incertesa a través del model. Però també és essencial a l'hora d'entrenar mètodes d'aprenentatge estadístic que incorporen simuladors físics en la seva funció de versemblança.

És possible obtenir un *emulador* eficaç i diferenciable mitjançant la generació d'un conjunt representatiu de parells de dades d'entrada-sortida simulats i utilitzar-los per formar un model d'aprenentatge estadístic. En aquest punt, abordem la qüestió de com es pot generar un conjunt de dades mentre s'executa el simulador el màxim de vegades possible.

7.4.1 Emulació activa

Per tal de construir un emulador eficient i computable de manera eficient, necessitem un conjunt de dades de parells d'entrada-sortida generades pel codi d'interès, al nostre cas el codi de transferència radiativa. El principal problema és que volem avaluar aquests models i generar simulacions el màxim nombre de vegades possible. A banda de que les simulacions poden ser costosos computacionalment, els algorismes basats en GPs poden, què són populars per emulació (O'Hagan, 2006), son molt ineficients computacionalment amb el nombre de punts d'entrenament. Per això, és important triar un conjunt reduït i representatiu de punts per tal d'avaluar el model en qüestió. Un enfocament comú a aquest problema és l'esquema de mostreig d'hipercub llatí (*Latin Hypercube Sampling*, LHS) (Audze, 1977) o simplement realitzant mostrejos aleatoris segons una distribució amb sentit físic. Aquests plantejaments, però, no tenen en compte el coneixement del comportament de la funció que es vol aprendre. A (Svendsen et al., 2020a), proporcionem una visió general de diferents mètodes desenvolupats per resoldre el problema de mostreig. En aquest treball proposem un mètode d'aprenentatge actiu (*active learning*, AL) (Settles, 2009) per resoldre aquest problema.

A partir d'un petit conjunt de dades de parells d'entrada-sortida simulada s'entrena un algorisme de regressió basat en GP, que es pot pensar com una versió inicial de l'emulador. Mitjançant l'estructura apresa per aquest emulador, es crea una funció d'adquisició que es pot optimitzar per trobar el punt més informatiu per tal d'avaluar el simulador. El nou punt s'afegeix al conjunt de dades i s'actualitza la funció de regressió, donant lloc a una nova funció d'adquisició. Aquest procediment es pot repetir fins a assolir un cost computacional màxim. Es tracta d'un mètode seqüencial que significa que utilitza els punts de dades generats anteriorment. Això contrasta amb mètodes no seqüencials com LHS, que calculen totes les ubicacions d'entrada en què executar el simulador una única volta. Utilitzar aquest mètode implica haver de conèixer prèviament la quantitat de punts necessaris per construir l'emulador i a més a més no es pot utilitzar conjuntament amb un conjunt de dades simulat ja existent.

7.4.2 Emulador actiu de processos gaussians multi-sortida

A (Svendsen et al., 2020a) argumentem que per trobar el punt d'entrada més informatiu per avaluar el model físic, la funció d'adquisició ha de combinar dos termes: 1) un *terme de diversitat* què, basat en nodes anteriors, reflecteix com és de buit l'espai d'entrada en un punt donat de l'espai, i 2) un *terme de geometria* que expressa la variabilitat de la funció subjacent en un punt determinat de l'espai. Argumentem que aquestes quantitats es deriven fàcilment d'una funció de regressió del GP com a 1) la variància predictiva i 2) el gradient de la mitjana predictiva, ambdós analítics. Per tant, basem el nostre esquema

d'aprenentatge actiu en processos gaussians amb múltiples eixides, construint maneres diferents de combinar les dues fonts d'informació sobre la funció subjacent.

7.4.3 Emulació activa d'un model de transferència radiativa

L'algorisme l'hem anomenat *Active multi-output Gaussian process emulator* (AMOGAPE) i l'hem aplicat en problemes sintètics i reals. En primer lloc hem fet una comparació exhaustiva del marc AMOGAPE amb un conjunt de mètodes d'última generació en un exemple joguina. Es van fer comparacions, tant amb mètodes seqüencials com ara mostreig aleatori, mostreig de seqüències de Sobol i un esquema de mostreig de mostreig LHS seqüencial, i mètodes no seqüencials com LHS i una graella determinista. Vam demostrar que AMOGAPE va ser capaç d'aprofitar la informació apresa de l'ajust de la funció subjacent per aconseguir una millor precisió utilitzant menys punts que els altres algorismes de mostreig.

Com a aplicació real, hem considerat el model de transferència radiativa PROSAIL. Variem dos dels paràmetres d'entrada més importants, a saber, l'índex d'àrea de les fulles (LAI) i el contingut de clorofil·la (Chl). Els paràmetres d'entrada restants es mantenen constants tal com s'explica a (Svendsen et al., 2020a). Hem comparat diferents funcions d'adquisició, basades en termes de diversitat i geometria, amb mostreig aleatori, provant els emuladors resultants en un conjunt de test de 50 000 punts simulats amb PROSAIL. Trobem que AMOGAPE supera el mostreig aleatori, necessitant menys punts d'entrenament per assolir una precisió predictiva millor.

7.5 Inferència amb maximització variacional i de l'esperança

Els models de transferència radiativa codifiquen el problema de recuperació del paràmetre en la direcció directa, *forward direction*, què hem estat estudiant. Hem vist com aquest model és útil per a la inversió directa mitjançant la generació de dades d'entrenament simulades per entrenar un algorisme de regressió per al mapejat en sentit invers. Un model forward també es pot utilitzar, però, per definir un model de probabilitat d'un enfocament probabilístic. En altres paraules, donat un vector de paràmetres físics, el model directe implementat pel codi RTM indueix una funció de versemblança, que vincula els paràmetres físics amb les reflexions observades. Aquesta interpretació obre noves opcions per incloure, per exemple, una distribució prèvia sobre paràmetres físics.

En aquest punt, abordem un problema general: volem aprendre la distribució dels paràmetres físics en lloc de proporcionar només una estimació puntual d'aquests paràmetres. Proporcionant un conjunt de dades de reflectàncies observades, el nostre objectiu és doble: aprendre la densitat marginal sobre paràmetres físics i obtenir una aproximació de la distribució posterior. Hem de tenir en compte que el a-posteriori representa un model invers probabilístic, és a dir, donada una reflectància, podem obtenir una predicció dels paràmetres i les mesures d'incertesa relacionades. A continuació, proposem i comparem dos enfocaments diferents que ens permet inferir paràmetres físics mitjançant un model de transferència. Un enfocament es basa en *variational autoencoders* (VAEs) (Kingma & Welling, 2013) i l'altre es basa en el *Monte Carlo expectation maximization* (MCEM) (Wei & Tanner, 1990). Mostrem que cada enfocament té pros i contres diferents.

7.5.1 Mètode d'inferència variacional

Els mètodes d'inferència variacional VAEs són algorismes d'aprenentatge no supervisat molt populars en la literatura de machine learning. Suposem un a-priori gaussià sobre un conjunt de variables latents i una probabilitat gaussiana on la mitjana i la covariància estan parametritzades per una xarxa neuronal (*neural network, NN*). Això fa que el a-posteriori siga intractable, un problema què es pot abordar amb la inferència variacional. El posterior variacional suposat és una altra gaussiana on la mitjana i la covariància estan parametritzades per una NN. Això permet una fàcil optimització de una fita de l'evidència respecte dels paràmetres NN que dona lloc a un codificador, el mapejat des de les observacions fins a valors latents i un descodificador que actua en sentit contrari. En la nostra aproximació substituïm la xarxa de descodificadors per un model de transferència radiativa de manera que el mapejat de la variable latent a l'observació esdevingui físicament significatiu. D'aquesta manera es produeix un a-priori sobre els paràmetres físics i un model invers format per la xarxa descodificadora.

7.5.2 Mètode de maximització de les expectatives de Monte Carlo

El mètode de maximització de l'esperança (*expectation maximization*, EM) és un mètode molt conegut per trobar iterativament estimacions (locals) de versemblances màximes. Funciona optimitzant el valor esperat de la funció *log-likelihood* dels paràmetres, respecte al posterior, mantenint els paràmetres fixats en l'expressió del posterior. D'aquesta manera s'obté un nou conjunt de paràmetres amb els quals es pot construir la *log-likelihood*, i així es continua fins a la convergència. Monte Carlo EM (MCEM) és el nom del mètode que utilitza el mostreig MC per estimar el valor esperat quan el veritable posterior és intractable com és el cas ací.

7.5.3 Resultats experimentals

Hem utilitzat diversos models avançats, experiment sintètics, així com el RTM PROSAIL. També hem demostrat que el mètode MCEM és més exigent computacionalment que l'enfocament VI. D'altra banda, el mètode VI proposat no pot tractar els a-posterioris multimodals. No obstant això, quan es realitza una inferència sobre el RTM PROSAIL, els mètodes han funcionat bé en general. El mètode MCEM va aconseguir obtenir una estimació lleugerament millor del veritable prior sobre les variables físiques.

7.6 Conclusions

Aquesta tesi ha presentat diferents maneres d'aplicar models probabilístics conjuntament amb coneixements de la física del problema per tal de desenvolupar algorismes d'aprenentatge automàtic adaptats a problemes específics de teledetecció. Una de les principals tendències de la literatura d'aprenentatge automàtic és desenvolupar algorismes que son agnòstics i només aprenen els patrons purament a partir de dades. Tot i que aquests algorismes donen bons resultats de predicció en general, poden produir prediccions que no siguen compatibles amb les lleis de la física. D'altra banda, hi ha un interès creixent en la intersecció entre els camps de la física i l'aprenentatge automàtic (Willard et al., 2020) i com es poden utilitzar mètodes d'un per millorar mètodes de l'altre. Entre les diferents direccions d'investigació en aquesta intersecció, aquesta tesi s'ha centrat a millorar la

modelització inversa codificant el coneixement a-priori en la regressió de l'aprenentatge automàtic i en la modelització directa amb l'emulació millorada de simuladors.

7.6.1 Sobre el modelatge invers amb processos gaussians

Hem començat mostrant com el problema invers de recuperació de paràmetres a partir de reflectàncies observades per satèl·lit, que sovint s'aborda amb la regressió GP (Camps-Valls et al., 2016), es pot afrontar de manera més eficient mitjançant una regressió profunda de GP. Els GP són freqüentment utilitzats en la teledetecció per diversos motius: es tracta d'una aproximació probabilística als problemes de regressió que condueixen a una expressió analítica de la predicció i la incertesa (Verrelst et al., 2013; Schneider et al., 2014). Això també permet la propagació eficaç d'errors des de les entrades a les sortides, com s'ha mostrat recentment a (Johnson et al., 2019).

Tanmateix, la regressió del procés gaussià té dos greus inconvenients: un pel que fa la impossibilitat d'aplicació a grans conjunts de dades, i per l'altra la incapacitat de modelar una estructura jeràrquica complexa mitjançant funcions de covariància estàndard. L'algorisme de GP d'inferència profundament estocàstica aplicat en aquest soluciona ambdós problemes; proporciona un cost lineal amb el nombre de dades d'entrenament i és capaç de modelar dades més complexes. Per al problema de recuperació de paràmetres biofísics a partir de dades de sensors infrarojos, hem demostrar que el model DGP supera els models sparse i estàndard de GP en termes de precisió predictiva i de qualitat de les estimacions d'incertesa. Al document complet (Svendsen et al., 2020b) també es mostra com els DGP es comparen favorablement amb les xarxes neurals d'última generació (Hieronymi et al., 2017) per a la recuperació de paràmetres de color de l'oceà mitjançant les dades òptiques del satèl·lit Sentinel-3.

A mesura que es desenvolupen sistemes híbrids que combinen models d'aprenentatge estadístic amb models físics, la millora de l'estimació de la incertesa esdevé més important per tal de realitzar una correcta propagació d'errors. A més, el flux creixent de dades de teledetecció disponibles requereix comptar amb algorismes d'aprenentatge automàtic escalables. Per aquestes raons, preveiem un futur augment de la regressió DGP en el camp de la recuperació de paràmetres biofísics.

7.6.2 Sobre la codificació del coneixement físic als procés gaussians

Primer hem desenvolupat un marc de GP per modelar conjuntament dades in situ i simulades sense deteriorar el rendiment predictiu de les dades *in situ*. Això s'ha aconseguit mitjançant la introducció d'un paràmetre de confiança que modela la variació relativa del soroll entre les fonts de dades. El fet d'ajustar els hiperparàmetres maximitzant la *leave-one-out likelihood* sobre les dades in situ garantia especialment que el mètode no s'ajustés massa a les dades simulades. L'esquema resultant va ser senzill i robust, i va millorar el rendiment predictiu de les dades in situ, especialment en els escenaris d'extrapolació. El marc s'ha estès a moltes fonts de dades i es pot utilitzar en dades generades amb diferents sensors o amb diferents simuladors.

A continuació, es van estudiar els models de força latent (Alvarez et al., 2009) per modelar la dinàmica de dades de humitat del sòl amb teledetecció basada en sensors de microones passives. Aquesta era una forma conceptualment diferent de codificar el coneixement expert ja que construeix el model de GP al voltant d'un conjunt d'assumpcions sobre el sistema físic actual. Més exactament, que les sortides es regeixen per una ODE

de primer ordre, els forçaments dels quals són processos gaussians. D'aquesta suposició es deriva un nucli de múltiples sortides amb hiperparàmetres amb sentit físic. Vam veure com les forces latents que s'aprenen durant la fase d'entrenament semblen els intervals de precipitació mesurats de forma independent. Aquest és un resultat prometedor, ja que la capacitat de monitorar l'impacte de les precipitacions en la humitat del sòl a partir de les mesures dels satèl·lits ajudaria a comprendre el cicle hidrològic de l'aigua. A més, les aplicacions dins de qualsevol sistema físic que es pugui mesurar amb teledetecció podria beneficiar-se d'aquest tipus de modelat si la dinàmica està prou descrita per una equació diferencial ordinària.

7.6.3 Sobre l'aprenentatge actiu per a l'emulació de RTMs

Els simuladors de processos físics solen ser costosos computacionalment, sovint en la mesura que impossibiliten aplicacions com l'anàlisi de sensibilitat. A més, els codis simuladors solen estar plens d'heurístics per fixar paràmetres interns de funcionament i d'operacions lògiques que fan que l'anàlisi de derivades sigui intractable matemàticament. Això fa que sigui difícil utilitzar aquests models en funcions de pèrdua que es tracten d'optimitzar. Els emuladors proporcionen una solució alternativa a aquests problemes, ja que les prediccions dels mètodes d'aprenentatge automàtic sovint consisteixen en operacions ràpides com les multiplicacions de matrius i es poden diferenciar fàcilment en el cas d'emprar GPs.

S'ha presentat en aquesta Tesi un enfocament d'aprenentatge actiu que incloïa informació sobre el gradient de la funció subjacent en la funció d'adquisició. La intuïció ha estat la de guiar el mostreig (o simulació) amb un criteri combinat que cerque regions poc explorades però alhora amb una alta variabilitat de funcions. Vam demostrar que el nostre mètode genera un emulador més precís per a PROSAIL que amb mètodes estàndard en la literatura. També vam comparar el nostre esquema de mostreig amb la seqüència Sobol i el mostreig llatí d'hipercubs (actualment utilitzat en moltes aplicacions), mostrant un rendiment millorat.

Entre els models de transferència radiativa existents, PROSAIL no és el més car computacionalment. El nostre mètode és especialment útil quan cada simulació requereix un temps considerable, com és el cas del MODTRAN RTM (Berk et al., 1987). En aquests casos, és crucial utilitzar la informació apresa sobre el RTM per avaluar-lo un mínim nombre de vegades.

7.6.4 Sobre la inferència aproximada dels RTM

Finalment, la Tesi ha abordat el problema de la inferència de les distribucions dels paràmetres amb una aproximació bayesiana aproximada al problema de la inversió. Aproximadament en el sentit que, juntament amb una funció de versemblança, definim un *família de priors* en lloc d'un prior específic que sol ser el cas de la inferència bayesiana. Fitem els paràmetres del a-priori (gaussià en aquest cas) i aprenem així la distribució conjunta del a-priori sobre els paràmetres físics d'interès. Se suposa que la versemblança és una gaussiana amb soroll baix, centrada al voltant de la sortida del PROSAIL, la qual cosa significa que el a posteriori actua com una mena de model invers, prenent reflectàncies observades i produint una distribució plausible de paràmetres biofísics.

Atès que el PROSAIL no és lineal, la inferència de la marginal likelihood és inviable. És per això que hem introduït dues maneres de realitzar una inferència aproximada de

manera estable, cadascuna amb els seus punts forts i febles. La primera està basada en l'autoencoder variacional (Kingma & Welling, 2013), on la xarxa de descodificadors es substitueix pel PROSAIL. Hem demostrat que aquest esquema d'inferència variacional és més ràpid d'entrenar. A més, una vegada aprés el posterior variacional, parametritzat per una xarxa neuronal, és molt ràpid d'avaluar, servint com a model invers ràpid. Aquest esquema d'inferència, però, no és adequat quan el posterior vertader és multimodal. Això no acaba sent un gran problema quan es realitza una inferència sobre el RTM.

El segon esquema d'inferència utilitzat va ser el de *Monte Carlo expectation maximization*. La fase d'entrenament consisteix en el mostreig del veritable posterior amb el mètode Hamilton Monte Carlo, fent que el mètode sigui més lent. Avaluar el posterior predictiu de les noves dades observades també implica el mostreig de Monte Carlo i, per tant, és molt més lent que l'alternativa VI. L'enfocament MCEM, però, pot modelar la multimodalitat en el vertader posterior que condueix a una millor convergència de l'algorisme quan aquest és el cas. Per tant, aquest mètode d'inferència condueix a estimacions lleugerament millors del veritable prior, com hem demostrat al fer inferència sobre PROSAIL.

Els problemes d'inversió són omnipresents en l'observació de la Terra, i hi ha molts models directes (*forward*) diferents que expressen el coneixement del domini de diversos sistemes físics. Una aproximació probabilística a aquests problemes de modelat directe i invers pot ser extremadament útil. Ens permet codificar coneixement expert en la funció de versemblança i escollir famílies plausibles de a-prioris. A diferència de aproximacions estàndard del ML en les que només es proporciona una estimació puntual, aquest tipus de marc descriu la distribució conjunta per paràmetres físics. És important assenyalar que hi ha literatura sobre l'acceleració dels mètodes de mostreig de Monte Carlo i mètodes que permeten aproximacions VI per modelar els posteriors multimodals (Martino & Elvira, 2017; Mescheder et al., 2017).

7.6.5 Treball futur

En el transcurs d'aquesta tesi, s'han publicat una sèrie de models d'aprenentatge estadístic de multi-fidelitat a la literatura (Pilania et al., 2017; Perdikaris et al., 2017). Aquest marc presenta una forma ideal de combinar dades reals i simulades, com es fa amb el JGP presentat en esta Tesi. Es preveuen moltes aplicacions d'aquest tipus de models a la literatura de teledetecció. L'esquema d'aprenentatge actiu amb GPs i el model de GP conjunt (JGP) presentats ací són dos exemples de mètodes on es podrien utilitzar altres tipus de models de GP. Com hem vist, el DGP supera els GP estàndard per a la estimació d'alguns paràmetres importants, i es podria pensar en utilitzar-lo al marc de JGP per a la modelització combinada.

Si bé aquesta tesi s'ha centrat en problemes en el camp de la teledetecció, hi ha molts camps de la ciència i l'enginyeria on els mètodes presentats serien útils. La majoria de ciències naturals utilitzen codis simuladors per analitzar diversos sistemes d'interès. Aquests simuladors es poden utilitzar per especificar models de probabilitat i realitzar inferències bayesianes, tal com s'ha mostrat en aquest treball, o per generar dades simulades per millorar la regressió de les dades *in situ*. Si els codis són costosos computacionalment, com sol passar, podrien beneficiar-se d'esquemes d'emulació parsimonioses com la que s'ha presentat ací. El treball presentat en aquesta tesi té, per tant, rellevància en molts altres camps de la ciència.

Bibliography

Adam, E., Mutanga, O., Abdel-Rahman, E. M., & Ismail, R. (2014). Estimating standing biomass in papyrus (cyperus papyrus l.) swamp: exploratory of in situ hyperspectral indices and random forest regression. *International Journal of Remote Sensing*, *35*, 693–714.

Alton, P., Ellis, R., Los, S., & North, P. (2007). Improved global simulations of gross primary product based on a separate and explicit treatment of diffuse and direct sunlight. *Journal of Geophysical Research: Atmospheres*, *112*.

Alvarez, M., Luengo, D., & Lawrence, N. D. (2009). Latent force models. In *Artificial Intelligence and Statistics* (pp. 9–16).

Alvarez, M. A., Rosasco, L., & Lawrence, N. D. (2011). Kernels for vector-valued functions: A review. *arXiv preprint arXiv:1106.6251*, .

Arfken, G. B., & Weber, H. J. (1999). Mathematical methods for physicists.

Audze, P. (1977). New approach to planning out of experiments. *Problems of dynamics and strengths*, *35*, 104–107.

Ba, Y., Zhao, G., & Kadambi, A. (2019). Blending diverse physical priors with neural networks. *arXiv preprint arXiv:1910.00201*, .

Babaeian, E., Sadeghi, M., Jones, S. B., Montzka, C., Vereecken, H., & Tuller, M. (2019). Ground, proximal, and satellite remote sensing of soil moisture. *Reviews of Geophysics*, *57*, 530–616.

Baret, F., Jacquemoud, S., Guyot, G., & Leprieur, C. (1992). Modeled analysis of the biophysical nature of spectral shifts and comparison with information content of broad bands. *Remote Sensing of Environment*, *41*, 133–142.

Bauer, M., van der Wilk, M., & Rasmussen, C. (2016). Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems* (pp. 1533–1541).

Berk, A., Bernstein, L. S., & Robertson, D. C. (1987). *MODTRAN: A moderate resolution model for LOWTRAN*. Technical Report SPECTRAL SCIENCES INC BURLINGTON MA.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, *112*, 859–877.

Bonilla, E. V., Chai, K. M., & Williams, C. (2008). Multi-task gaussian process prediction. In *Advances in neural information processing systems* (pp. 153–160).

Bratley, P., & Fox, B. (1988). Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator. *ACM Transactions on Mathematical Software (TOMS)*, *14*, 88–100.

Brocca, L., Moramarco, T., Melone, F., & Wagner, W. (2013). A new method for rainfall estimation through soil moisture observations. *Geophysical Research Letters*, *40*, 853–858.

Broge, N. H., & Leblanc, E. (2001). Comparing prediction power and stability of broadband and hyperspectral vegetation indices for estimation of green leaf area index and canopy chlorophyll density. *Remote sensing of environment*, *76*, 156–172.

Campos-Taberner, M., García-Haro, F. J., Camps-Valls, G., Grau-Muedra, G., Nutini, F., Crema, A., & Boschetti, M. (2016). Multitemporal and multiresolution leaf area index retrieval for operational local rice crop monitoring. *Remote Sensing of Environment*, *187*, 102–118.

Camps-Valls, G., Gómez-Chova, L., Laparra, V., Martino, L., Mateo-García, G., Muñoz-Marí, J., Svendsen, D. H., & Verrelst, J. (2020). Statistical biophysical parameter retrieval and emulation with gaussian processes. In *Data Handling in Science and Technology* (pp. 333–368). Elsevier volume 32.

Camps-Valls, G., Tuia, D., Gmez-Chova, L., Jimnez, S., & Malo, J. (2011). *Remote Sensing Image Processing*. (1st ed.). Morgan & Claypool Publishers.

Camps-Valls, G., Verrelst, J., Muñoz-Marí, J., Laparra, V., Mateo-Jimenez, F., & Gomez-Dans, J. (2016). A survey on gaussian processes for earth-observation data analysis: A comprehensive investigation. *IEEE Geoscience and Remote Sensing Magazine*, *4*, 58–78.

Coccia, G., Siemann, A. L., Pan, M., & Wood, E. F. (2015). Creating consistent datasets by combining remotely-sensed data and land surface model estimates through bayesian uncertainty post-processing: The case of land surface temperature from hirs. *Remote Sensing of Environment*, *170*, 290–305.

Damianou, A., & Lawrence, N. (2013). Deep Gaussian processes. In *Artificial Intelligence and Statistics* (pp. 207–215).

Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P., & Karpatne, A. (2020). Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. In *Proceedings of the 2020 SIAM International Conference on Data Mining* (pp. 532–540). SIAM.

Delworth, T. L., & Manabe, S. (1988). The influence of potential evaporation on the variabilities of simulated soil wetness and climate. *Journal of Climate*, *1*, 523–547.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*, 1–22.

Faurtyot, T., & Baret, F. (1997). Vegetation water and dry matter contents estimated from top-of-the-atmosphere reflectance data: A simulation study. *Remote Sensing of Environment*, *61*, 34–45.

Geneva, N., & Zabaras, N. (2020). Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, *403*, 109056.

Gewali, U. B., Monteiro, S. T., & Saber, E. (2019). Gaussian processes for vegetation parameter estimation from hyperspectral data with limited ground truth. *Remote Sensing*, *11*, 1614.

Gómez-Dans, J. L., Lewis, P. E., & Disney, M. (2016). Efficient emulation of radiative transfer codes using gaussian processes and application to land surface parameter inferences. *Remote Sensing*, *8*, 119.

Gustau Camps-Valls, J. R. M. R., Dino Sejdinovic (2019). A perspective on gaussian processes for earth observation. *National Science Review*, *6*, 616–618. doi:https://doi.org/10.1093/nsr/nwz028.

Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence* (pp. 282–290). AUAI Press.

Hieronymi, M., Mueller, D., & Doerffer, R. (2017). The OLCI Neural Network Swarm (ONNS): A bio-geo-optical algorithm for open ocean and coastal waters. *Frontiers in Marine Science*, *4*, 140.

Jacquemoud, S., Verhoef, W., Baret, F., Bacour, C., Zarco-Tejada, P., Asner, G., François, C., & Ustin, S. (2009). PROSPECT+ SAIL models: A review of use for vegetation characterization. *Remote sensing of environment*, *113*, S56–S66.

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., & Kumar, V. (2019). Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining* (pp. 558–566). SIAM.

Johnson, J. E., Laparra, V., & Camps-Valls, G. (2019). Accounting for input noise in gaussian process parameter retrieval. *IEEE Geoscience and Remote Sensing Letters*, .

Jordan, C. F. (1969). Derivation of leaf-area index from quality of light on the forest floor. *Ecology*, *50*, 663–666.

Karpatne, A., Watkins, W., Read, J., & Kumar, V. (2017). Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, .

Kattge, J., Bönisch, G., Díaz, S., Lavorel, S., Prentice, I. C., Leadley, P., Tautenhahn, S., Werner, G. D., Aakala, T., Abedi, M. et al. (). Try plant trait database–enhanced coverage and open access. *Global change biology*, .

Kattge, J., Diaz, S., Lavorel, S., Prentice, I. C., Leadley, P., Bönisch, G., Garnier, E., Westoby, M., Reich, P. B., Wright, I. J. et al. (2011). Try–a global database of plant traits. *Global change biology*, *17*, 2905–2935.

Kingma, D. P., & Ba, J. (2015). ADAM: a method for stochastic optimization. In *Inrernational Conference on Learning Representations* (pp. 1–15).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, .

Kogan, F. N. (1995). Application of vegetation index and brightness temperature for drought detection. *Advances in space research*, *15*, 91–100.

Kondratyev, K. Y., & Varotsos, C. (2002). Remote sensing and global tropospheric ozone observed dynamics. *International Journal of Remote Sensing*, *23*, 159–178.

Lagerloef, G. S., Swift, C. T., & Le Vine, D. M. (1995). Sea surface salinity: The next remote sensing challenge. *Oceanography*, *8*, 44–50.

Laio, F., Porporato, A., Ridolfi, L., & Rodriguez-Iturbe, I. (2001). Plants in water-controlled ecosystems: active role in hydrologic processes and response to water stress: Ii. probabilistic soil moisture dynamics. *Advances in water resources*, *24*, 707–723.

Lawrence, N. D., Sanguinetti, G., & Rattray, M. (2007). Modelling transcriptional regulation using gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 785–792).

Leen, G., Peltonen, J., & Kaski, S. (2012). Focused multi-task learning in a gaussian process framework. *Machine learning*, *89*, 157–182.

Li, X., Liu, X., Liu, M., & Wu, L. (2014). Random forest algorithm and regional applications of spectral inversion model for estimating canopy nitrogen concentration in rice. *J. Remote Sens*, *18*, 934–945.

Liang, S. (2004). *Quantitative Remote Sensing of Land Surfaces*. New York: John Wiley & Sons.

Llorente, F., Martino, L., Delgado, D., & Lopez-Santiago, J. (2020). Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *arXiv:2005.08334*, (pp. 1–58).

Ma, C., Li, X., Notarnicola, C., Wang, S., & Wang, W. (2017). Uncertainty quantification of soil moisture estimations based on a bayesian probabilistic inversion. *IEEE Transactions on Geoscience and Remote Sensing*, *55*, 3194–3207.

MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, *168*, 133–166.

Malmgren-Hansen, D., Laparra, V., Nielsen, A. A., & Camps-Valls, G. (2019). Statistical retrieval of atmospheric profiles with deep convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, *158*, 231–240.

Martino, L., & Elvira, V. (2017). Metropolis sampling. *Wiley StatsRef: Statistics Reference Online*, (pp. 1–15).

Martino, L., Elvira, V., Luengo, D., & Corander, J. (2017). Layered adaptive importance sampling. *Statistics and Computing*, *27*, 599–623.

Martino, L., Luengo, D., & Míguez, J. (2018). *Independent Random Sampling Methods*. springer.

Mateo-García, G., Adsuara, J. E., Pérez-Suay, A., & Gómez-Chova, L. (2019). Convolutional long short-term memory network for multitemporal cloud detection over landmarks. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (pp. 210–213). IEEE.

Mateo-García, G., Gómez-Chova, L., & Camps-Valls, G. (2017). Convolutional neural networks for multispectral image cloud masking. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 2255–2258). IEEE.

Mateo-Sanchis, A., Piles, M., Muñoz-Marí, J., Adsuara, J. E., Pérez-Suay, A., & Camps-Valls, G. (2019). Synergistic integration of optical and microwave satellite data for crop yield estimation. *Remote Sensing of Environment*, *234*, 111460.

McCarthy, A., Rodriguez, B., & Mincholé, A. (2017). Variational inference over non-differentiable cardiac simulators using bayesian optimization. *arXiv preprint arXiv:1712.03353*, .

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*, 239–245. doi:10.1080/00401706.1979.10489755.

Melkumyan, A., & Ramos, F. (2011). Multi-kernel gaussian processes. In *Twenty-second international joint conference on artificial intelligence*.

Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning* (pp. 2391–2400).

Morales-Alvarez, P., Pérez-Suay, A., Molina, R., & Camps-Valls, G. (2017). Remote sensing image classification with large-scale gaussian processes. *IEEE Transactions on Geoscience and Remote Sensing*, *56*, 1103–1114.

Moreno-Martinez, A., Camps-Valls, G., Kattge, J., Robinson, N., Reichstein, M., van Bodegom, P., Kramer, K., Cornelissen, J., Reich, P., Bahn, M., Niinemets, U., Penuelas, J., Craine, J., Cerabolini, B., Minden, V., Laughlin, D., Sack, L., Allred, B., Baraloto, C., Byun, C., Soudzilovskaia, N., & Running, S. (2018). A methodology to derive global maps of leaf traits using remote sensing and climate data. *Remote Sensing of Environment*, *218*, 69–88. doi:https://doi.org/10.1016/j.rse.2018.09.006.

Mott, H. (2007). *Remote Sensing with Polarimetric Radar*. New York: John Wiley & Sons.

Neal, R. M. et al. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, *2*, 2.

Neath, R. C. (2013). On convergence properties of the Monte Carlo EM algorithm. In *Advances in modern statistical theory and applications: A festschrift in honor of Morris L. Eaton* (pp. 43–62).

O'Hagan, A. (2006). Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, *91*, 1290–1300.

Peñuelas, J., Gamon, J., Fredeen, A., Merino, J., & Field, C. (1994). Reflectance indices associated with physiological changes in nitrogen- and water-limited sunflower leaves. *Remote Sens Environ*, *48*, 135–146.

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., & Karniadakis, G. E. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *473*, 20160751.

Pilania, G., Gubernatis, J. E., & Lookman, T. (2017). Multi-fidelity machine learning models for accurate bandgap predictions of solids. *Computational Materials Science*, *129*, 156–163.

Piles, M., van der Schalie, R., Gruber, A., Muñoz-Marí, J., Camps-Valls, G., Mateo-Sanchís, A., Dorigo, W., & de Jeu, R. (2018). Global estimation of soil moisture persistence with L and C-Band microwave sensors. In *2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 8259–8262).

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, .

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning* (pp. 63–71). Springer.

Rivera, J. P., Verrelst, J., Gómez-Dans, J., Muñoz-Marí, J., Moreno, J., & Camps-Valls, G. (2015). An emulator toolbox to approximate radiative transfer models with statistical learning. *Remote Sensing*, *7*, 9347–9370.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, (pp. 400–407).

Robert, C., & Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Rosenstein, M. T., Marx, Z., Kaelbling, L. P., & Dietterich, T. G. (2005). To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning* (pp. 1–4). volume 898.

Rouse, J., Haas, R., Schell, J., & Deering, D. (1974). Monitoring vegetation systems in the great plains with erts. *NASA special publication*, *351*, 309.

Ruescas, A. B., Mateo-Garcia, G., Camps-Valls, G., & Hieronymi, M. (2018). Retrieval of case 2 water quality parameters with machine learning. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 124–127). IEEE.

Salimbeni, H., & Deisenroth, M. (2017). Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 4588–4599).

Sanchez, N., Martínez-Fernández, J., Scaini, A., & Perez-Gutierrez, C. (2012). Validation of the smos l2 soil moisture data in the remedhus network (spain). *IEEE Transactions on Geoscience and Remote Sensing*, *50*, 1602–1611.

Schneider, S., Murphy, R. J., & Melkumyan, A. (2014). Evaluating the performance of a new classifier–the gp-oad: A comparison with existing methods for classifying rock type and mineralogy from hyperspectral imagery. *ISPRS journal of photogrammetry and remote sensing*, *98*, 145–156.

Schölkopf, B., Smola, A. J., Bach, F. et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Settles, B. (2009). *Active learning literature survey*. Technical Report University of Wisconsin-Madison Department of Computer Sciences.

Shaw, G., & Manolakis, D. (2002). Signal processing for hyperspectral image exploitation. *IEEE Signal Proc. Magazine*, *50*, 12–16.

Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems* (pp. 1257–1264).

Sobol, I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical modelling and computational experiments*, *1*, 407–414.

Spreen, G., Kaleschke, L., & Heygster, G. (2008). Sea ice remote sensing using amsr-e 89-ghz channels. *Journal of Geophysical Research: Oceans*, *113*.

Sundarajan, S., & Keerthi, S. (2001). Predictive approaches for choosing hyper-parameters in gaussian process. *Neural Computation*, *13*, 1103–1118.

Svendsen, D. H., Martino, L., Campos-Taberner, M., García-Haro, F. J., & Camps-Valls, G. (2017). Joint gaussian processes for biophysical parameter retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, *56*, 1718–1727.

Svendsen, D. H., Martino, L., & Camps-Valls, G. (2020a). Active emulation of computer codes with gaussian processes–application to remote sensing. *Pattern Recognition*, *100*, 107103.

Svendsen, D. H., Morales-Álvarez, P., Ruescas, A. B., Molina, R., & Camps-Valls, G. (2020b). Deep gaussian processes for biogeophysical parameter retrieval and model inversion. *ISPRS Journal of Photogrammetry and Remote Sensing*, *166*, 68–81.

Tournier, B., Blumstein, D., Cayla, F., , & Chalon, G. (2002). IASI level 0 and 1 processing algorithms description. In *Proc. of ISTCXII Conference*.

Tramontana, G., Jung, M., Camps-Valls, G., Ichii, K., Ráduly, B., Reichstein, M., Schwalm, C. R., Arain, M. A., Cescatti, A., Kiely, G. et al. (2016). Predicting carbon dioxide and energy fluxes across global fluxnet sites with regression algorithms. *Biogeosciences Discussions*, .

Ustin, S. (2004). *Remote Sensing for Natural Resource Management and Environmental Monitoring. Manual of Remote Sensing, Volume 4*. New York: John Wiley & Sons.

Verrelst, J., Rivera, J., Moreno, J., & Camps-Valls, G. (2013). Gaussian processes uncertainty estimates in experimental Sentinel-2 LAI and leaf chlorophyll content retrieval. *ISPRS Journal of Photogrammetry and Remote Sensing*, *86*, 157–167.

Vicent, J., Alonso, L., Martino, L., Sabater, N., Verrelst, J., Camps-Valls, G., & Moreno, J. (2019). Gradient-based automatic look-up table generator for radiative transfer models. *IEEE Transactions on Geoscience and Remote Sensing*, *57*, 1040–1048.

Vicent, J., Verrelst, J., Sabater, N., Alonso, L., Rivera-Caicedo, J. P., Martino, L., Muñoz-Marí, J., & Moreno, J. (2020). Comparative analysis of atmospheric radiative transfer models using the atmospheric look-up table generator (alg) toolbox (version 2.0). *Geoscientific Model Development*, *13*, 1945–1957.

Wang, B.-C. (2008). *Digital Signal Processing Techniques and Applications in Radar Image Processing*. New York: John Wiley & Sons.

Wei, G. C. G., & Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, *85*, 699–704.

Van der Wilk, M., Rasmussen, C. E., & Hensman, J. (2017). Convolutional gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 2849–2858).

Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, .

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*, 37–52.

Yang, Z., Wu, J.-L., & Xiao, H. (2019). Enforcing deterministic constraints on generative adversarial networks for emulating physical systems. *arXiv preprint arXiv:1911.06671*, .

Zhang, Q., Xiao, X., Braswell, B., Linder, E., Baret, F., & Moore III, B. (2005). Estimating light absorption by chlorophyll, leaf and canopy in a deciduous broadleaf forest using modis data and a radiative transfer model. *Remote Sensing of Environment*, *99*, 357–371.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*, 8–36.

Appendix: Scientific Publications

Publication I

Svendsen, D.H., Morales-Álvarez, P., Ruescas, A.B., Molina, R. and Camps-Valls, G., 2020. Deep Gaussian processes for biogeophysical parameter retrieval and model inversion. *ISPRS Journal of Photogrammetry and Remote Sensing, 166*, pp.68-81.

Q1: Computer Science Applications, Q1: Computers in Earth Sciences, Q1: Atomic and Molecular Physics, and Optics, Q1: Geography, Planning and Development, IF = 6.94

Publication II

Svendsen, D.H., Martino, L., Campos-Taberner, M., García-Haro, F.J. and Camps-Valls, G., 2017. Joint Gaussian processes for biophysical parameter retrieval. *IEEE Transactions on Geoscience and Remote Sensing, 56(3)*, pp.1718-1727.

Q1: Earth and Planetary Sciences, Q1: Electrical and Electronic Engineering, IF = 5.63

Publication III

Svendsen, D.H., Martino, L. and Camps-Valls, G., 2020. Active emulation of computer codes with Gaussian processes–Application to remote sensing. *Pattern Recognition, 100*, p.107103.

Q1: Artificial Intelligence, Q1: Computer Vision and Pattern Recognition, Q1: Signal Processing, Q1: Software, IF = 5.90

ISPRS Journal of Photogrammetry and Remote Sensing 166 (2020) 68–81

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Deep Gaussian processes for biogeophysical parameter retrieval and model inversion

Daniel Heestermans Svendsen[a,*], Pablo Morales-Álvarez[b], Ana Belen Ruescas[a], Rafael Molina[b], Gustau Camps-Valls[a]

[a] Image Processing Lab (IPL), Universitat de València, C/ Cat. José Beltrán, 2., 46980 Paterna, Spain
[b] Department of Computer Science and Artificial Intelligence, University of Granada, 18010 Granada, Spain

ARTICLE INFO

Keywords:
Model inversion
Statistical retrieval
Deep Gaussian Processes
Machine learning
Moisture
Temperature
Chlorophyll content
Inorganic suspended matter
Coloured dissolved matter
Infrared sounder
IASI
Sentinels
Copernicus programme

ABSTRACT

Parameter retrieval and model inversion are key problems in remote sensing and Earth observation. Currently, different approximations exist: a direct, yet costly, inversion of radiative transfer models (RTMs); the statistical inversion with *in situ* data that often results in problems with extrapolation outside the study area; and the most widely adopted hybrid modeling by which statistical models, mostly nonlinear and non-parametric machine learning algorithms, are applied to invert RTM simulations. We will focus on the latter. Among the different existing algorithms, in the last decade kernel based methods, and Gaussian Processes (GPs) in particular, have provided useful and informative solutions to such RTM inversion problems. This is in large part due to the confidence intervals they provide, and their predictive accuracy. However, RTMs are very complex, highly nonlinear, and typically hierarchical models, so that very often a single (shallow) GP model cannot capture complex feature relations for inversion. This motivates the use of deeper hierarchical architectures, while still preserving the desirable properties of GPs. This paper introduces the use of deep Gaussian Processes (DGPs) for bio-geo-physical model inversion. Unlike shallow GP models, DGPs account for complicated (modular, hierarchical) processes, provide an efficient solution that scales well to big datasets, and improve prediction accuracy over their single layer counterpart. In the experimental section, we provide empirical evidence of performance for the estimation of surface temperature and dew point temperature from infrared sounding data, as well as for the prediction of chlorophyll content, inorganic suspended matter, and coloured dissolved matter from multispectral data acquired by the Sentinel-3 OLCI sensor. The presented methodology allows for more expressive forms of GPs in big remote sensing model inversion problems.

1. Introduction

Estimating variables and bio-geophysical parameters of interest from remote sensing images is a central problem in Earth observation (Liang, 2008; Rodgers, 2000; Gómez-Chova et al., 2011). This is usually addressed through a very challenging *model inversion problem*, which involves dealing with complex nonlinear input–output relations. In addition, very often, the goal is to invert *metamodels*, that is, combinations of submodels that are coupled together. In remote sensing, radiative transfer models (RTMs) describe the processes which occur at different scales (e.g. at leaf, canopy and atmospheric levels) with different complexities. The overall process is thus complicated, nonlinear and hierarchical, with different sources of uncertainty propagating through the system.

The inversion of such highly complex models has been attempted through several strategies. One standard approach consists on running a reasonable number of RTM simulations which generates the so called look-up tables (LUTs). Then, for a new input observation, one assigns the most similar parameter in the LUT. A second, more direct approach involves the direct physics-based inversion, which results in complex optimization problems. An alternative *hybrid approach* comes from the use of statistical approaches to perform the inversion using the LUT simulations. A review of approaches can be found in (Verrelst et al., 2012; Gómez-Chova et al., 2011). In recent years, the remote sensing community has turned to this type of statistical *hybrid* approaches for model inversion (Gómez-Chova et al., 2011), mainly because of efficiency, versatility and the interesting balance between its data driven and physics-aware nature (Verrelst et al., 2015).

* Corresponding author.
E-mail addresses: daniel.svendsen@uv.es (D.H. Svendsen), pablomorales@decsai.ugr.es (P. Morales-Álvarez), ana.b.ruescas@uv.es (A.B. Ruescas), rms@decsai.ugr.es (R. Molina), gustau.camps@uv.es (G. Camps-Valls).

https://doi.org/10.1016/j.isprsjprs.2020.04.014
Received 7 October 2019; Received in revised form 14 April 2020; Accepted 23 April 2020

Approximating arbitrary nonlinear functions from data is a solid field of machine learning where many successful methods are available. Data-driven statistical learning algorithms have attained outstanding results in the estimation of climate variables and related geo-physical parameters at local and global scales (Camps-Valls and Bruzzone, 2009; Gómez-Chova et al., 2011). These algorithms avoid complicated assumptions and provide flexible non-parametric models that fit the observations using large heterogeneous data. The fact is that a plethora of regression algorithms have been used. There exist traditional models such as random forests (Tramontana et al., 2016; Jung et al., 2017) and standard feed-forward neural networks (Blackwell, 2005; Blackwell et al., 2008; Camps-Valls et al., 2012) as well as convolutional neural networks (Malmgren-Hansen et al., 2019; Ma et al., 2019).

In the last decade, more emphasis has been put on kernel methods in general (Camps-Valls and Bruzzone, 2009; Rojo-Álvarez et al., 2018), and Gaussian Processes (GPs) in particular. There is a considerable amount of reasons for this. Firstly, GPs constitute a probabilistic treatment of regression problems leading to an analytical expression for the predictive uncertainty which is an attractive feature (Verrelst et al., 2013; Schneider et al., 2014). This also allows for effective error propagation from the inputs to the outputs as has recently been shown in (Johnson et al., 2019). Furthermore, GPs are not pure black box models because, through the use and design of appropriate covariance functions, one can include prior knowledge about the signal characteristics (e.g. nonstationarity, heteroscedasticity, etc.). The covariance hyperparameters are learned (inferred) from data so that the model is interpretable. For instance, by using the automatic relevance determination (ARD) covariance function (Verrelst et al., 2016), an automatic feature ranking can be derived from the trained model, thus leading to a explanatory model. These theoretical and practical advantages have recently translated to a wider adoption by the geoscience and remote sensing community in many applications and products, such as the spatialization of in situ measurements and upscaling of carbon, energy and water fluxes (Jung et al., 2017). Gaussian Processes have provided very good results for retrieval in all Earth science domains, be it land and vegetation parameter retrieval (Furfaro et al., 2006; Rivera-Caicedo et al., 2017; Camps-Valls et al., 2018; Gustau Camps-Valls and Sejdinovic, 2019), ocean and water bodies modeling (Ruescas et al., 2018a; Sarkar et al., 2019), cryosphere ice sheet modeling and process emulation (Wernecke et al., 2019), or atmospheric parameter retrieval (Camps-Valls et al., 2012).

Despite being successful in many different applications, standard GPs have two important shortcomings we want to highlight:

- *Computational cost.* A standard GP, which stores and uses all the data at once, exhibits a high computational cost. These GPs scale cubically with the number of data points when training, and quadratically when doing prediction. This hampers their adoption in applications which involve more than just a few thousand input points.
- *Expressiveness.* GPs are shallow models,[1] so while accurate and flexible, their expressive power is limited when dealing with hierarchical structures. This is even worse due to the (ab) use of standard kernel functions like the exponentiated quadratic family (e.g. the RBF kernel is infinite-differentiable and tends to oversmooth functions).

The first limitation is typically addressed through sparse GPs (Snelson and Ghahramani, 2006), which have already been used in remote sensing applications (Morales-Alvarez et al., 2017). In order to additionally tackle the second limitation, in this paper we introduce the use of Deep Gaussian Process (DGP) (Salimbeni and Deisenroth, 2017) to the field of remote sensing for the first time. A DGP is a cascaded and hierarchical model that captures more complex data structures, while still being able to scale well to millions of points. Our proposal is not incidental: the complexity of the processes involved in geosciences and remote sensing leads to highly hierarchical and modular models to be inverted. This calls for the application of the most innovative available techniques as shown in the following example. Fig. 1 compares the use of a standard GP and deep GP to model a hurricane structure. It becomes clear that, unlike GPs, the DGP can cope with the whirl structure efficiently by combining different latent functions hierarchically.

DGPs were originally introduced in (Titsias and Lawrence, 2010; Damianou and Lawrence, 2013), and further analyzed in (Damianou, 2015). In (Svendsen et al., 2018), we outlined the potential use of DGPs for surface level dew point temperature retrieval from sounding data. In this paper we extend that work in several ways: 1) we focus the analysis on large scale remote sensing problems, aiming for a complete treatment of the two aforementioned standard GP shortcomings; 2) provide a deeper formalization and more intuitive insight on the model for the practitioner; 3) give more empirical evidences of performance in ocean and land parameter retrieval applications, and using different sensory data (optical sensors and microwave sounders); and 4) assess accuracy and robustness to sample sizes and problem dimensionality versus both standard and sparse implementations of GPs.

In short, this work exposes the DGP methodology to the remote sensing community for the first time and for a wide range of applications. The proposed DGP appears to be an excellent approach for model inversion. Moreover, sticking to the GP framework is very convenient. GPs are based on a solid Bayesian formalism and inherit all properties of a probabilistic treatment: possibility to derive not only point-wise predictions but also confidence intervals, perform error quantification and uncertainty propagation easily, and optimize hyperparameters by log-likelihood maximization.

The remainder of the paper is organized as follows. Section 2 establishes notation, reviews the probabilistic modeling and inference of GP and sparse GP, and presents the deep GP model - mathematical details on modeling, inference, and prediction are provided in Appendices A, B, and C. Section 3 provides the experimental results. We illustrate performance in prediction of surface temperature and dew point temperature (related to relative humidity) from superspectral infrared sounding data (Aires, 2002; Siméoni et al., 1997; Huang et al., 1992); as well as for the estimation of predicting chlorophyll content, inorganic suspended matter, and coloured dissolved organic matter from simulated multispectral data acquired by Sentinel-3 OLCI sensor. Finally, Section 4 concludes the paper with summarizing remarks.

2. Probabilistic model and inference

In this section we provide a brief and graphical introduction to modeling and inference for Gaussian Processes (GP) and Deep Gaussian Processes (DGP) in supervised regression problems. We explain and graphically show the hierarchical structure of DGPs, and also explain how both GPs and DGPs, make use of sparse approximations to perform inference tasks. Mathematical details are deferred to appendices.

Gaussian Processes are non-parametric probabilistic state-of-the-art models for functions, and are successfully used in supervised learning. In geostatistics, GPs for regression is usually referred to as *kriging*. The main strength of GPs is their accurate uncertainty quantification, which is a consequence of its sound Bayesian formulation, yielding well-calibrated predictions and confidence intervals (Rasmussen and Williams, 2006; Damianou, 2015).

More specifically, for input–output data $\{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i=1}^n$, a GP models the underlying dependence with latent variables $\{f_i = f(\mathbf{x}_i) \in \mathbb{R}\}_{i=1}^n$ that jointly follow a Gaussian distribution $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$. The kernel matrix $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j}$ encodes the properties (e.g. smoothness) of the modeled functions. The most popular

[1] It can be shown that, in the limit and under some mild assumptions, a GP corresponds to a single-hidden layer neural network with infinite neurons (Neal, 1996).

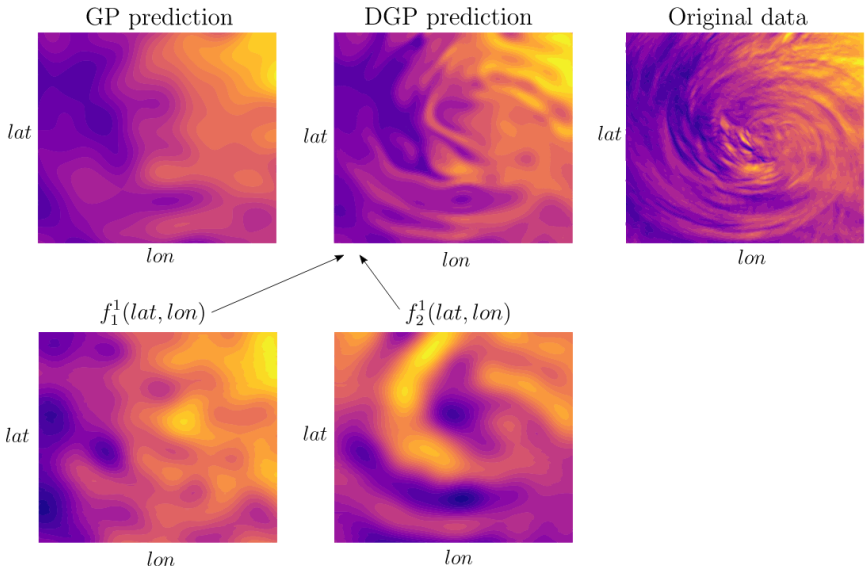


Fig. 1. Example of shallow versus deep GPs. Modeling a hurricane field from the coordinates using 1000 randomly selected training points in both cases. The GP prediction [top left] is too blurry and does not capture the whirl data structure (the scale of feature relations changes along the hurricane ridges). The DGP model [top middle] uses only two latent functions in its first layer. The first latent function f_1 captures lower frequencies [bottom left] –similarly to the GP map– and the f_2 [bottom middle] focuses on the hurricane structure, while their combination leads to an overall predictive function [top middle] that better approximates the observation [top right].

standard kernel is the squared exponential one (or RBF), which is given by $k(\mathbf{x}, \mathbf{y}) = \gamma \cdot \exp(-\|\mathbf{x} - \mathbf{y}\|^2/(2\sigma^2))$, with γ (variance) and σ (length-scale) called the kernel hyperparameters. Finally, in regression problems, the observation model of the outputs y_i given the latent variables f_i is usually defined by the Gaussian $p(y_i|f_i, \rho^2) = \mathcal{N}(y_i|f_i, \rho^2)$. The variance ρ^2 is estimated during the training step, along with the kernel hyperparameters, by maximizing the marginal likelihood of the observed data.

Since the Gaussian prior $p(\mathbf{f})$ is conjugate to the Gaussian observation model, one can integrate out $\mathbf{f}$ and compute the marginal likelihood $p(\mathbf{y})$ and the posterior $p(\mathbf{f}|\mathbf{y})$ in closed form (parameters are omitted for simplicity) (Rasmussen and Williams, 2006). However, this requires inverting the $n \times n$ matrix $(\mathbf{K} + \rho^2\mathbf{I})$, which scales cubically, $O(n^3)$ where n is the number of training data points. This constraint makes GP prohibitive for large scale applications, with $n = 10^4$ usually being considered the practical limit (Morales-Álvarez et al., 2017). Here, sparse GP approximations become the preferred pathway to scale the desirable properties of GPs to larger datasets (Snelson and Ghahramani, 2006; Hensman et al., 2013; Bauer et al., 2016; Morales-Álvarez et al., 2017), and they will be reviewed in Section 2.1. Interestingly, we will see that DGPs preserve the scalability of sparse GP approximations (while achieving a higher expressiveness).

Additionally, GPs are limited by the expressiveness of the kernel function. Ideally, complex kernels could be tailored for different applications (Rasmussen and Williams, 2006). However, this is usually unfeasible in practice, as it requires a thorough application-specific knowledge. Moreover, it usually comes with a large amount of hyperparameters to estimate, which may cause overfitting. As a result, standard general-purpose kernels are normally considered in practice. Alternatively, DGPs allow for modeling very complex data through a hierarchy of GPs that only use simple kernels with few hyperparameters as building blocks (like the aforementioned RBF one, which will be used here). Fig. 2 provides an intuition on this, and DGPs will be introduced in Section 2.2.

2.1. Sparse GP approximations

In the last years, many different sparse GP approximations have been introduced in order to cope with increasingly large datasets (Snelson and Ghahramani, 2006; Hensman et al., 2013; Bauer et al., 2016; Morales-Álvarez et al., 2017). Most of them resort to the notion of *inducing points*, a reduced set of $m \ll n$ latent variables which the inference is based on. More specifically, these inducing points $\mathbf{u} = (u_1, ..., u_m)$ are GP realizations at the *inducing locations* $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_m\} \subset \mathbb{R}^d$, just like $\mathbf{f}$ is at the inputs $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$. All these sparse methods are grouped in two big categories, depending on where exactly the approximation takes place: in the model definition or in the inference procedure (Bauer et al., 2016). Both types of sparse GP will be compared against the deep GP in the experiments.

In the first group, the *Fully Independent Training Conditional* (FITC) (Snelson and Ghahramani, 2006) is the most popular approach. It uses the inducing points to approximate the GP model, and then marginalizes them out and perform exact inference. This yields a reduced $O(nm^2)$ computational complexity (linear in the dataset size). Mathematical details for FITC are included in Appendix A.

In the second group, the *Scalable Variational Gaussian Process* (SVGP) (Hensman et al., 2013) is one of the most widespread methods. It maintains the exact GP model, and uses the inducing points to introduce the approximation in the inference process through variational inference (Blei et al., 2017). The mathematical details are included in Appendix B (which is devoted to DGPs because, as we explain in next paragraph, SVGP is equivalent to DGP with one layer). Since SVGP does not modify the original model, it is less prone to overfitting. However, if the posterior distribution is not well approximated within the variational scheme, its performance might become poorer. Therefore, both groups of methods are complementary, and in the machine learning community none of them is considered to consistently outperform the other (Bauer et al., 2016). An advantage of SVGP over FITC is its factorization in mini-batches, which allows for even greater scalability. In this case, the computational cost is $O(n_b m^2)$, with n_b the mini-batch size.

Interestingly, the second paradigm (exact model plus approximate inference) has proven to translate well to hierarchical concatenations of GPs, yielding the inference process for DGPs that is presented in next section. This justifies that SVGP will be equivalently referred to as DGP ($L = 1$) hereafter. This is also graphically depicted in Fig. 3. Moreover, as explained before, Fig. 3 shows that $\mathbf{u}$ is integrated out in FITC after the model approximation, whereas it is maintained in DGP ($L = 1$), where an (approximate) posterior distribution is calculated for it. As a general summary, Table 1 shows the main differences between the four GP-based methods that will be used in this work (standard GP, sparse GP FITC, sparse GP SVGP, and deep GP), which are also represented in Fig. 3.

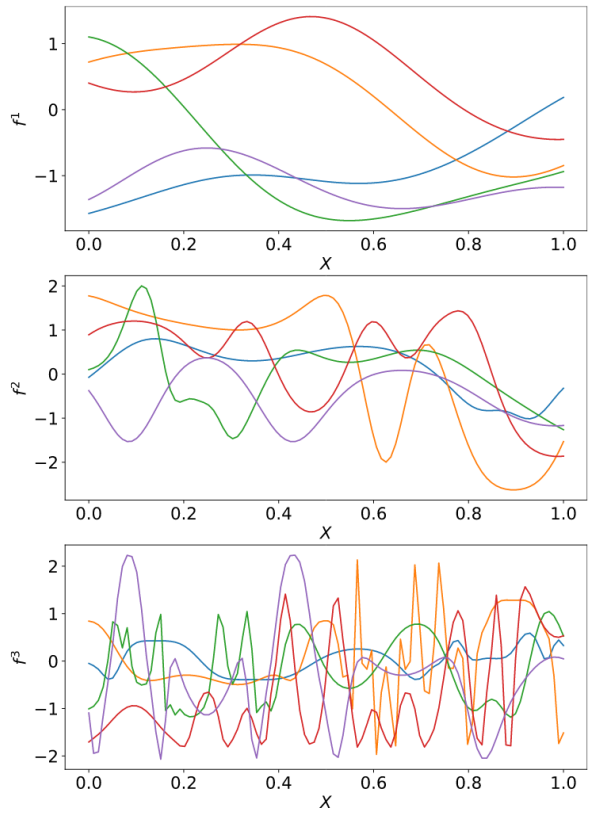


Fig. 2. Five random samples from a 1-dimensional DGP with three layers and one hidden unit per layer. Each function sample uses the function of the same color in the previous plot as input, except the function samples of the top plot (L = 1) which use the actual values of x as input. Every layer is endowed with a standard RBF kernel. This produces very smooth functions in the first layer (i.e. a shallow GP, top plot). However, the concatenation of such simple GPs produces increasingly complex functions (middle and bottom plots, 2-layer and 3-layer DGPs respectively). In particular, notice that DGP-3 captures sophisticated patterns that combine flat regions with high-variability ones, which cannot be described by stationary kernels. These ideas are behind the superiority of DGPs in Fig. 1.

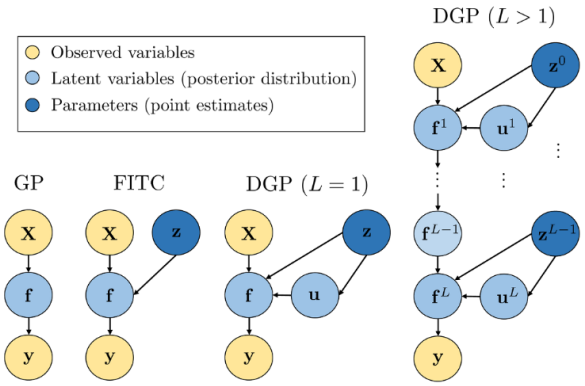


Fig. 3. Graphical representation of the four GP-based models used in this work. The color indicates whether a variable is observed or must be estimated. In the latter case, the intensity of the color represents the type of estimation: either through a posterior distribution (light), or a point value (dark). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.2. Deep Gaussian processes

In standard (single-layer) GPs, the output of the GP is directly used to model the observed response $\mathbf{y}$. However, this output could be used to define the input locations of another GP. If this is repeated L times, we obtain a hierarchy of GPs that is known as a Deep Gaussian Process (DGP) with L layers. This is analogous to the structure of deep neural networks, which are a cascade of generalized linear models (Damianou, 2015, Chapter 6). Intuitively, this stacked composition will be able to capture more complex patterns in the training data, recall Fig. 2. See also Fig. 3 for a graphical depiction of the DGP model.

DGPs were first introduced in (Damianou and Lawrence, 2013), where the authors performed approximate variational inference analytically. In order to achieve this tractability, in each layer they define a set of latent variables which end up inducing *independence across layers* in the posterior distribution approximation. This uncorrelated posterior fails to express the complexity of the deep model, and is not realistic in practice. To overcome this problem, we present the recent inference procedure of (Salimbeni and Deisenroth, 2017), which keeps a strong conditional dependence in the posterior by marginalizing out the aforementioned set of latent variables. In exchange, analytic tractability is sacrificed. However, we will see that the structure of the posterior allows one to efficiently sample from it and use Monte Carlo approximations. As will be justified in Appendix B, this approach is called *Doubly Stochastic Variational Inference* (Salimbeni and Deisenroth, 2017).

DGPs can be used for regression by placing a Gaussian likelihood after the last layer. For notation simplicity, in the sequel the dimensions of the hidden layers will be fixed to one (this can be generalized straightforwardly, see both approaches (Damianou and Lawrence, 2013; Salimbeni and Deisenroth, 2017)). But exact inference in DGPs is intractable (not only computationally expensive as in GPs), as it involves integrating out latent variables that are used as inputs for the next layer (i.e. they appear inside a complex kernel matrix). To overcome this, m inducing points $\mathbf{u}^l$ at inducing locations $\mathbf{z}^{l-1}$ are introduced at each layer l. Interestingly, we will see that this sparse formulation also makes DGP scale well to large datasets, transferring the scalability of (shallow) sparse GP approximations like SVGP up to hierarchical structures.

For observed $\{\mathbf{X}, \mathbf{y}\}$, the regression joint DGP model is

$$p(\mathbf{y}, \{\mathbf{f}^l, \mathbf{u}^l\}_1^L) = p(\mathbf{y}|\mathbf{f}^L) \prod_{l=1}^{L} p(\mathbf{f}^l|\mathbf{u}^l;\mathbf{f}^{l-1}, \mathbf{z}^{l-1}) p(\mathbf{u}^l;\mathbf{z}^{l-1}). \tag{1}$$

Here, $\mathbf{f}^0 = \mathbf{X}$, and each factor in the product is the joint distribution over $(\mathbf{f}^l, \mathbf{u}^l)$ of a GP in the inputs $(\mathbf{f}^{l-1}, \mathbf{z}^{l-1})$, but rewritten with the conditional probability given $\mathbf{u}^l$. Notice that a semicolon is used to specify the inputs of the GP. The rightmost plot in Fig. 3 shows a graphical representation of the described model.

The Doubly Stochastic Variational Inference for this model is detailed in Appendix B, see also (Salimbeni and Deisenroth, 2017). Basically, assuming that the inducing points are enough to summarize the information contained in the training data, the model log-likelihood can be lower bounded by a quantity (called the Evidence Lower Bound, ELBO) that factorizes across data points. This allows for training in mini-batches, just as in SVGP, which makes DGPs scalable to large datasets. Finally, the prediction of the DGPs for a new test data point is included in Appendix C.

2.3. Implementation and practicalities

Several implementations of DGPs are currently available. In our experiments, we used the code integrated within GPflow (a GP framework built on top of Tensorflow), which is publicly available at https://github.com/ICL-SML/Doubly-Stochastic-DGP. We also used GPflow to train the standard GP and both sparse GP approaches: FITC and SVGP

Table 1
Summary of the main differences between the four GP-based models used in this work. VI = Variational Inference.

	GP	FITC	DGP ($L = 1$)	DGP ($L > 1$)
Model	Exact	Approx.	Exact	Exact
Inference	Exact	Exact	Approx. (VI)	Approx. (VI)
Depth	Shallow	Shallow	Shallow	Deep
Training cost	$\mathcal{O}(n^3)$	$\mathcal{O}(nm^2)$	$\mathcal{O}(n_b m^2)$	$\mathcal{O}\left(n_b m^2 \sum_{l=1}^{L} D^l\right)$
References	Rasmussen and Williams (2006)	Snelson and Ghahramani (2006)	Hensman et al. (2013), Salimbeni and Deisenroth (2017)	Salimbeni and Deisenroth (2017)

(equivalently, DGP with $L = 1$). In addition, for the sake of reproducibility, we provide illustrative code and demos in a Jupyter notebook at http://isp.uv.es/dgp/. The used data is available upon request.

3. Experimental results

The problem of translating radiances to state parameters is challenging because of its intrinsic high nonlinearity and underdetermination. We consider two such relevant remote sensing problems which together span both land and ocean application, namely (1) predicting surface level temperature and dew point temperature from infrared sounding data, and (2) predicting chlorophyll content, inorganic suspended matter and coloured dissolved matter from S3-OLCI data. Both problems involve inverting a model using large datasets of different sample size and dimensionality. In the first problem we compare DGPs with (shallow) standard and sparse GPs, highlighting the benefit of going deep in the GP setting. We also illustrate the predictive power of the DGP as a function of depth and data scale. The second problem aims at comparing the proposed model to another state-of-the-art method in a challenging real application. Specifically, we compare the performance of a DGP architecture with that of a state-of-the-art neural network method described in (Hieronymi et al., 2017).

3.1. Surface temperature and moisture from infrared sounders

Temperature and water vapour are essential meteorological parameters for weather forecasting studies. Observations from high spectral resolution infrared sounding instruments on board satellites provide unprecedented accuracy of temperature and water vapour profiles. However, it is not trivial to retrieve the full information content from radiation measurements. Accordingly, improved retrieval algorithms are desirable to achieve optimal performance for existing and future infrared sounding instrumentation. The use of MetOp data observations has an important impact on several Numerical Weather prediction (NWP) forecasts. The Infrared Atmospheric Sounding Interferometer (IASI) sensor is implemented on the MetOp satellite series. In particular, IASI collects rich spectral information to derive temperature and moisture (EUMETSAT, 2014; Tournier et al., 2002). EUMETSAT, NOAA, NASA and other operational agencies are continuously developing product processing facilities to obtain L2 products from infrared hyperspectral radiance instruments, such as IASI, AIRS or the upcoming MTG-IRS. Nonlinear statistical retrieval methods, and in particular kernel machines and Gaussian processes, have proven useful in retrieval of temperature and dew point temperature (humidity) recently (Camps-Valls et al., 2012; Laparra et al., 2015; Laparra et al., 2017). Here we explore the use of deep Gaussian processes to retrieve surface temperature and moisture from IASI data.

3.1.1. Data collection and pre-processing

The IASI instrument scans the Earth at an altitude of, approximately, 820 km. The instrument measures in the infrared part of the electromagnetic spectrum (specifically between wavenumbers 645 cm^{-1} and 2760 cm^{-1}, i.e. at wavelengths from 15.5 μm to 3.62 μm) at a horizontal resolution of 12 km over a swath width of, approximately, 2200 km. It obtains a global coverage of the Earth's surface every 12 h, representing 7 orbits in a sun-synchronous mid-morning orbit, and the data obtained from it are used for meteorological models. Each orbit consists of approximately 92 000 samples collected at a spatial resolution of 0.5 degrees. This represents more than one million high dimensionality samples to be processed each day.

Obtaining all the products provided by IASI with classical methods requires an enormous computational load. Each original sample has 8461 spectral bands, but following previous recommendations (Camps-Valls et al., 2012) we performed feature selection removing the most noisy bands and keeping 4699. Then we projected the data into the top 50 principal components to combat the risk of overfitting when working with such a high dimensional space. Each pixel is matched with the temperature and dew point temperature at surface level estimated using the European Center for Medium-Range Weather Forecasts (ECMWF) model.

3.1.2. Experimental setup

We employed the data collected in 14 consecutive orbits within the same day, namely the 1st of January 2013, by the IASI sensor. We carried out two different experiments within this application. The first one analyzes how the training data size influences the accuracy in all the GP-related methods, including different depths for the DGP. The second one compares the performance when partitioning the data according to geographical information, such a biome and climate zones. Additionally, it analyzes the quality of the predictive uncertainty. In the following we refer to these two separate experiments as *Experiment-1* and *Experiment-2* respectively.

Experiment-1: In order to analyze the effect of the training data size, we randomly shuffle the data, and select training sets of sizes 10 000, 50 000, 140 000, and 250 000, and a testing set of 20 000. The root mean squared error (RMSE) that will be reported is the average over five repetitions of the experiment. The compared models are named as follows:

DGP1–4: DGP described in Section 2.2 with 1–4 layers and 300 inducing inputs per layer. The number of hidden units per layer is 5. Recall that DGP1 is equivalent to the sparse GP method SVGP, and the computational cost of DGP is $O(n_b m^2(D^1+...+D^L))$.

FITC: Introduced in Section 2.1. Along with SVGP, it is the most popular sparse GP approximation. The RBF kernel is used, and the code

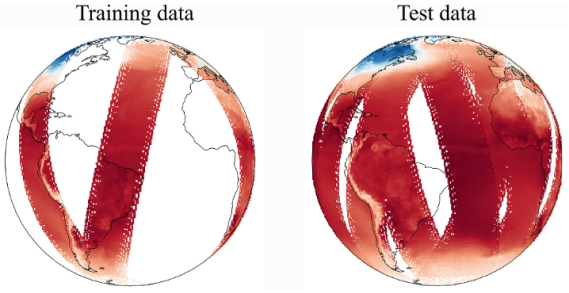


Fig. 4. Orbit-wise partition into training and test set used for model comparison when partitioning according to different biomes and climatic zones.

is taken from GPflow.[2] The cost of training scales like $O(nm^2)$, and the number inducing points is 300.

GP-10 K: A standard GP using 10 000 training points is provided as a baseline. Recall that this is the limit of a standard GP in practice, since it scales like $O(n^3)$. Again, the RBF kernel and the GPflow library are used.

Experiment-2: Out of the 14 available orbits we choose 11 for test data, and partition it according to: climatic zone, the dominant biome at the location of each data point, latitude, and whether a data point is located at land or at sea. We then selected training data from the remaining 3 orbits (see Fig. 4) and trained one model from each family: A standard, a sparse and a deep Gaussian Process. The models with their sizes of training data were respectively: A standard GP with 10 000, a FITC with 250 000 and a 3-layer DGP 250 000 data points, the data size reflecting the scalability of each training procedure. Comparing the predictions on the test-dataset of ~10^6 points, we also perform an analysis of the provided estimates of predictive uncertainty.

3.1.3. Experimental results

The results of *Experiment-1* are summarized in Fig. 5. We immediately see that there is a clear difference in RMSE between the shallow (GP-10 K, FITC, DGP1) and the improved deep models (DGP2-4). As intuitively expected, the performance of all models increases with additional training data. In this particular problem, it appears that the majority of additional complex structure is learned by going from 10^4 to 5×10^4 data points. As the DGP1 and FITC models are only approximations of the standard GP, it is to be expected that they perform worse when training on the same amount of data as the GP-10 K. Nevertheless, when allowed to leverage more data, their fit improves and outperforms the GP-10 K. It is not clear which of the two approximations is superior, as it varies with the number of training data. This agrees well with the literature, where this has been shown to depend on the data at hand (Bauer et al., 2016). The fact that single-layer approximations can outperform a standard GP when given enough training data underlines the importance of a model which is able to handle large-scale data. We can see from the results that the DGP both handles large datasets but also allows for higher model complexity and thus a better fit of the data. From observing the performance of DGPs with different numbers of layers, we can see that DGPs take advantage of their hierarchical structure and achieve lower RMSE with increasing depth. There is a considerable improvement when going from 2 to 3 layers, whereas the effect of going from 3 to 4 layers seems less significant.

We now turn to *Experiment-2* for the comparison of the three different GP types, trained according to what their computational cost allows them: We compare the GP-10 K with a FITC and a 3-layer DGP model both trained on 250 000 data points. Comparing the predictions on the ~10^6 test points (obtained from the 11 orbits shown in Fig. 4)

[2] https://github.com/GPflow.

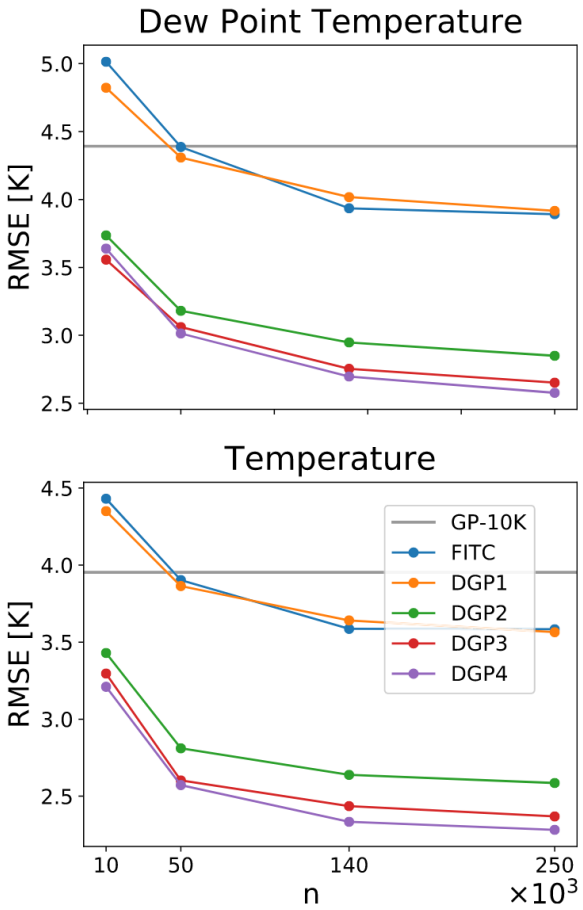


Fig. 5. Performance of the compared methods as a function of the training set size for the surface dew point temperature (top) and temperature (bottom) variables. The plots share the abscissa. The RMSE of the Deep Gaussian Processes decreases with increasing depth. The DGPs outperform the FITC which performs similarly to the GP-10K.

with the ground truth, we can analyze the quality of the predictive uncertainty provided by the models. Each model provides, for a given test point y^*, a Gaussian predictive distribution with a mean $\mu(x^*)$ and a variance $\sigma^2(x^*)$ - see Appendix A for the expression for the GP and FITC models, and Appendix C for the expression for DGPs. Scaling residuals of the predictive mean by the predictive standard deviation we obtain a variable $\zeta^* = \frac{\mu(\mathbf{x}^*) - y^*}{\sigma(\mathbf{x}^*)}$ which according to the model should follow a $\mathcal{N}(0, 1)$ distribution. Scaling the residuals from prediction on the 11 test orbits in this way, we can make a Kernel Density Estimation (KDE) to analyze their empirical distribution. The modes of the empirical distributions shown in Fig. 6 are shifted to the left, indicating a general underestimation in the predictive models. If a model yields too low uncertainties in general (over-confidence), the scaled residuals will become very large and their empirical distributions would have long tails. Conversely, if the model yields too high uncertainties as a rule, the corresponding empirical distribution would be narrowly centered around 0. It can be seen from Fig. 6 that the scaled residuals of the DGP model follow a $\mathcal{N}(0, 1)$ distribution closer than those of the other models, implying that the DGP does the best job of determining the predictive uncertainty correctly. This superior estimate of uncertainty may be due to its higher hierarchical representation capability, accounting for more complex structure in the data. In practice, this implies improved estimates of how certain the model is about its results

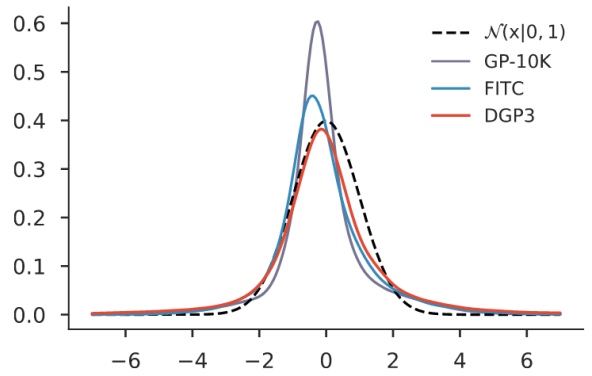


Fig. 6. KDE of residuals normalized by predictive standard deviation, which according to the model should be standard normal distributed. The 3-layer DGP avoids the underestimation seen in the other models, and provides better estimates of predictive uncertainty.

Table 2
Mean absolute error (in [K]) and standard error for each model, for predicting surface temperature using different partitions of the test results: land-vs-ocean, climatic zones, per latitude and per biome over land. F and BF are short for Forest and Broadleaf Forest, respectively. Details on climatic and biome classes are given in Appendix D.

Land/Ocean	GP-10 K	FITC	DGP3
Land	4.95 (0.02)	4.96 (0.02)	**3.75** (0.02)
Ocean	2.10 (0.01)	1.96 (0.01)	**1.59** (0.01)
Climatic zone			
Tropical	1.90 (0.03)	**1.84** (0.03)	1.88 (0.03)
Arid	5.17 (0.07)	5.02 (0.06)	**4.72** (0.07)
Temperate	4.30 (0.18)	4.60 (0.19)	**3.54** (0.18)
Cold	6.13 (0.20)	6.21 (0.20)	**3.75** (0.16)
Polar	6.81 (0.06)	6.89 (0.06)	**4.60** (0.07)
Latitude			
[+40, +60]	2.09 (0.01)	2.01 (0.01)	**1.84** (0.01)
[+20, +40]	2.52 (0.02)	2.22 (0.01)	**2.20** (0.02)
[0, +20]	1.64 (0.01)	**1.45** (0.01)	1.58 (0.01)
[−20, 0]	2.23 (0.01)	1.89 (0.01)	**1.70** (0.01)
[−40, −20]	3.96 (0.03)	4.04 (0.03)	**3.26** (0.03)
[−60, −40]	5.22 (0.03)	5.31 (0.03)	**3.88** (0.02)
Biome			
Needleleaf F	6.61 (0.11)	6.80 (0.11)	**3.38** (0.08)
Evergreen BF	1.98 (0.03)	1.94 (0.03)	**1.71** (0.02)
Deciduous BF	5.02 (0.10)	4.83 (0.10)	**2.86** (0.09)
Mixed forest	7.45 (0.07)	7.48 (0.07)	**5.01** (0.08)
Shrublands	6.44 (0.06)	5.66 (0.06)	**4.16** (0.05)
Savannas	2.90 (0.04)	2.83 (0.04)	**2.19** (0.03)
Grasslands	6.27 (0.06)	6.42 (0.06)	**5.17** (0.07)
Croplands	4.87 (0.05)	4.90 (0.05)	**3.44** (0.04)
Total	5.20	5.16	**4.17**

when performing parameter retrieval.

For the problem of predicting surface temperature, the mean absolute error (MAE) of the trained GP, FITC and DGP3 are shown in Table 2 for different partitions of the test data (and the whole test set in the last row). For the majority of partitions, DGP3 outperforms the other models, showing that the positive effects of deeper structures are not particular to a type of data, but extend across various meaningful partitions. FITC in turn, being capable of leveraging more data as it does not suffer from the cubic computational cost of the GP, outperforms GP which is trained on what is widely considered to be its upper limit for the number of training points (~10^4). In tropical regions of the northern hemisphere, DGP3 performs slightly less accurately than FITC, as seen from the partition into climatic zones and different latitudes. Differences in model assumptions and training schemes among machine learning algorithms can cause the models to focus on slightly different parts of the data. It can be concluded, however, that DGPs in general provide much better performance than their shallow counterparts, both due to their ability to leverage large amounts of data and to model more complex data than their shallow counterparts.

3.2. Ocean color parameters from optical sensors

Since the first remote sensing images of the ocean were taken, ocean color retrievals have been produced regularly with more or less accuracy, depending on the target parameter, in different regions of the planet and for several water types. The water quality variables reported as able to be estimated by remote sensing are: concentration of inorganic suspended matter (ISM), turbidity, colored dissolved organic matter (CDOM), concentration of chlorophyll-a (Chl-a), occurrence of surface accumulating algal blooms, concentration of phycocyanin, and Secchi depth, e.g. (Morel and Prieur, 1977; Bukata et al., 1995; Dekker et al., 2001). Research was initially more focused on open ocean or Case-1 waters - where optical properties are determined mainly by the phytoplankton contribution - later with further development of algorithms for more complex or Case-2 waters (Prieur and Sathyendranath, 2019). The development and validation of water quality algorithms, many of them empirically developed and implemented using *in situ* data from very specific locations, are the main topic of many of the published investigations. The development of algorithms that do not require extensive *in situ* sampling for training has become an aim in remote sensing of water quality (Kallio, 2006). For that reason, new databases that combine *in situ* and derived simulated data with radiative transfer models are becoming the training source of semi-analytic and machine learning approaches, like neural networks or Bayesian methods (Doerffer and Schiller, 2007; Hieronymi et al., 2017; Bayesian methodology for inverting satellite ocean-color data, 2015). The experiment carried out here uses one of those recently developed databases, designed within the framework of a European Space Agency (ESA) project called Case 2 eXtreme (C2X), in order to provide a database for the training and validation of a neural net approach (Hieronymi et al., 2017). A subset of this dataset has been already used to test five machine learning approaches, including simple Gaussian processes, for the determination of the three basic ocean colour parameters (Ruescas et al., 2018b).

3.2.1. Data collection and pre-processing

Within the framework of the Case 2 eXtreme (C2X) project (Hieronymi et al., 2016), in-water radiative transfer simulations for Sentinel 3-Ocean and Land Instrument (OLCI) were carried out with the commercial software Hydrolight (Mobley and Sundman, 2013). For more detail on the source of the simulations see (Hieronymi et al., 2015; Kraseman et al., 2016). In the C2X project, the results of the simulations were grouped into five subcategories: Case 1, Case 2 Absorbing (C2A), Case 2 Absorbing-Extreme (C2AX), Case 2 Scattering (C2S) and Case 2 Scattering-Extreme (C2SX), depending on the optical type of water with dominance of absorbing substances (more related to Chl and CDOM) or scattering particles (ISM) in several magnitudes (Hieronymi et al., 2016). Each subcategory consists of 20 000 individual combinations of concentration of water constituents, inherent optical properties (IOPs), and sun positions. One part of the S3-OLCI simulated dataset is put aside for validation purposes, with more than 4000 spectra per sub-category reserved exclusively for that. The C2X dataset contains simulations in 21 bands, from which a subset of 11 bands is used here for water quality parameter estimation as in (Hieronymi et al., 2017). This large dataset was used for the training and testing of the S3-OLCI Neural Network Swarm (ONNS) in-water processor

(Hieronymi et al., 2017). ONNS is the result of blending various NN algorithms, each optimized for a specific water type, covering the largest possible variability of water properties including oligotrophic and extreme waters. Results from the DGP approach will be compared with the ones achieved by ONNS as part of the validation process.

3.2.2. Experimental setup

In the present experiment we have selected all data available for the five categories included in the C2X dataset. In total we have 10^5 records that we use to train and test the DGP models. As already mentioned, 11 out of the 21 S3-OLCI wavebands are selected as inputs, from 400 to 885 nm. The $a_{CDOM}(440)$ nm absorption coefficient, using all subgroups (C1, C2A, C2AX, C2S and C2SX), has a range between 0.098 and 20 m^{-1}; while the Chl-a content range rises from 0.03 until 200 mg m^{-3}; inorganic suspended matter (ISM) ranges from 0.02 to more than 100 g m^{-3}. This means that the dataset incorporates a broad range of optical water combinations, making it an effective representation of global ocean and coastal waters including extreme cases. The purpose of this experiment is to generate the three most popular remote sensing water quality variables (CDOM, Chl-a and ISM) per water category (C1, C2A, C2AX, C2S and C2SX), using DGPs. Other works published on the matter have already used GPs to calculate the three parameters with a subset of the C2X dataset (Ruescas et al., 2018b; Ruescas et al., 2018c) with promising results. Subsets of the data had to be used in the aforementioned works, as a standard GP cannot leverage data in the order of magnitude presented in the present paper. The DGP model was trained and validated using the same data, 8×10^4 training and 2×10^4 test data points respectvely, as the ONNS (Hieronymi et al., 2017). The results of the ONNS will be used as a source for comparison, that is, we compare our results with state-of-the-art deep learning methods globally accepted in the OC community.

3.2.3. Experimental results

A 3-layer DGP with 500 inducing points and 5 GPs in each hidden layer is trained. Adding more layers was found not to increase performance significantly. This amount of inducing points is frequently used in the GP literature (see e.g. (Shi et al., 2019)), and is set to deal with the higher complexity of the C2X dataset. Table 3 shows the comparison of the RMSE between DGP and ONNS dividing the test set by water type, and the total (bottom row). The highlighted results are: compared to ONNS, CDOM results improve in the extreme absorbing and scattering waters, which also affects the total RMSE (DGP 0.115 mg m^{-3} against ONNS 0.202 mg m^{-3}). ISM results improve in scattering waters, staying on the same range of error for the other water types, which also translates into almost a factor 3 improvement with the total dataset (DGP 5.296 against ONNS 15.134 g m^{-3}). However, the most impressive results are observed for Chl, where more than a factor 3 improvement in the RMSE can be observed for all water cases. (Table 4).

We visualize the behaviour of measured against predicted values in Fig. 7. In this figure the actual values (x-axis) vs. the DGP predictions (y-axis) are compared by variable and water type, in a similar fashion as was done by (Hieronymi et al., 2017) with the ONNS results, with the exception of the non-log scale of our figure. In the following we make references to model predictions in regions of low numerical value which are better appreciated in the log-scale version of the figure located in Appendix E. Summarizing the results by water quality parameter:

- CDOM: High uncertainties and distribution dispersion in Case 1 and scattering waters (C2S(X)) for very low CDOM values (<0.2 m^{-1}). To separate the CDOM from suspended sediments using the absorption signal seems not to be easy. The correlation improves for absorbing waters (C2A(X)) for all values, with good uncertainty ranges for high values in C2A waters, with a gradual increase for the CDOM range higher than 15 m^{-1} in extreme cases.
- ISM: Shows almost a perfect correlation for C2S and C2SX, which are the scattering waters where the main component are suspended sediments and non-algal particles. CDOM dominated waters (C2A and C2AX) are not expected to have high non-organic suspended sediment content, which gives less relevance to the more dispersed and less accurate results in these water types. Some saturation is observed in absorbing waters for very low values < 0.1 g m^{-3}, which is better appreciated in Fig. 8, as well as for C1 waters, where dispersion is in general higher; however, it shows lower uncertainty values. This result is in line with the ONNS results in which "the retrieval performance is less skilled if the optical signal of minerals is weak due to low mineral concentrations as is the case in oligotrophic waters (C1)" (Hieronymi et al., 2017).
- Chl: Despite the lower values of uncertainty, there seems to be some overestimation in the minimum Chl values (concentrations < 1.0 mg m^{-3}) of all five water types. General bias and dispersion is higher in the C2A and C2AX cases. This is an indication of the complexity of the separation of Chl and CDOM for these types of waters. Uncertainty is incremented with high concentration values in all five water types (> 100 mg m^{-3}), increasing the dispersion of the data points considerably in C2SX water with Chl values higher than (> 150 mg m^{-3}), with a clear underestimation of the parameter. In any case, in nature, cases of extreme ISM and Chl concentration are rare.

Considering in the analysis the different water types, Case 1 waters shows quite good results for Chl values (< 100 mg m^{-3}), with an increase in the uncertainty and dispersion of the data with higher concentrations. ISM uncertainties are generally low but dispersion and bias are high all through the range. CDOM detection can be problematic and tend to underestimation for values (< 0.5 m^{-1}). However, ISM and CDOM are not elements usually found in oligotrophic waters, where Chl is the main contributor to the colour of the water. In C2A and C2AX absorbing waters, Chl values < 0.5 mg m^{-3} and > 100 mg m^{-3} will be difficult to quantify properly. Lower Chl values would be probably underestimated in C2AX. High Chl concentrations show higher dispersion and uncertainties in both types of water. The ISM distribution is these absorbing waters looks quite good, and uncertainties keep generally low. Dispersion is higher with higher ISM values. CDOM retrievals, however, show a quite good fit to the 1:1 line in C2AX, with an increase of the uncertainty with the increase of CDOM absorption. In C2A waters there is more dispersion around the 1:1 line and more variability in the uncertainty range in values (< 1.0 m^{-1}). In high scattering waters (C2S and C2SX), there is underestimation in the quantification of high Chl values. CDOM distribution shows cases of over or underestimation depending on the range and water type, with medium to low uncertainties found in low CDOM values. ISM fit to the 1:1 line is very good for both scattering water types, showing C2SX water higher uncertainties in the lower and higher ranges of ISM.

A comparison with the results of (Hieronymi et al., 2017), can be made, taking into account the several differences between both approaches. On the one hand, the most remarkable fact is that ONNS is a so-called swarm of neural nets designed from and for several water

Table 3
Comparison of RMSE between a 3-layer DGP and the ONNS dividing the test set by watertype, and without dividing the test set (bottom row).

	CDOM		ISM		Chl	
	DGP	ONNS	DGP	ONNS	DGP	ONNS
C1	0.0584	**0.0174**	0.1393	**0.0856**	**2.7913**	10.2577
C2A	0.0324	**0.0234**	0.1648	**0.116**	**2.2126**	10.5276
C2AX	**0.2429**	0.4356	0.2156	**0.1429**	**2.6765**	10.3555
C2S	**0.0200**	0.029	**0.7784**	1.4917	**2.6668**	11.2224
C2SX	**0.0270**	0.0971	**12.476**	35.689	**2.5617**	16.7635
Total	**0.115**	0.202	**5.296**	15.134	**2.594**	11.914

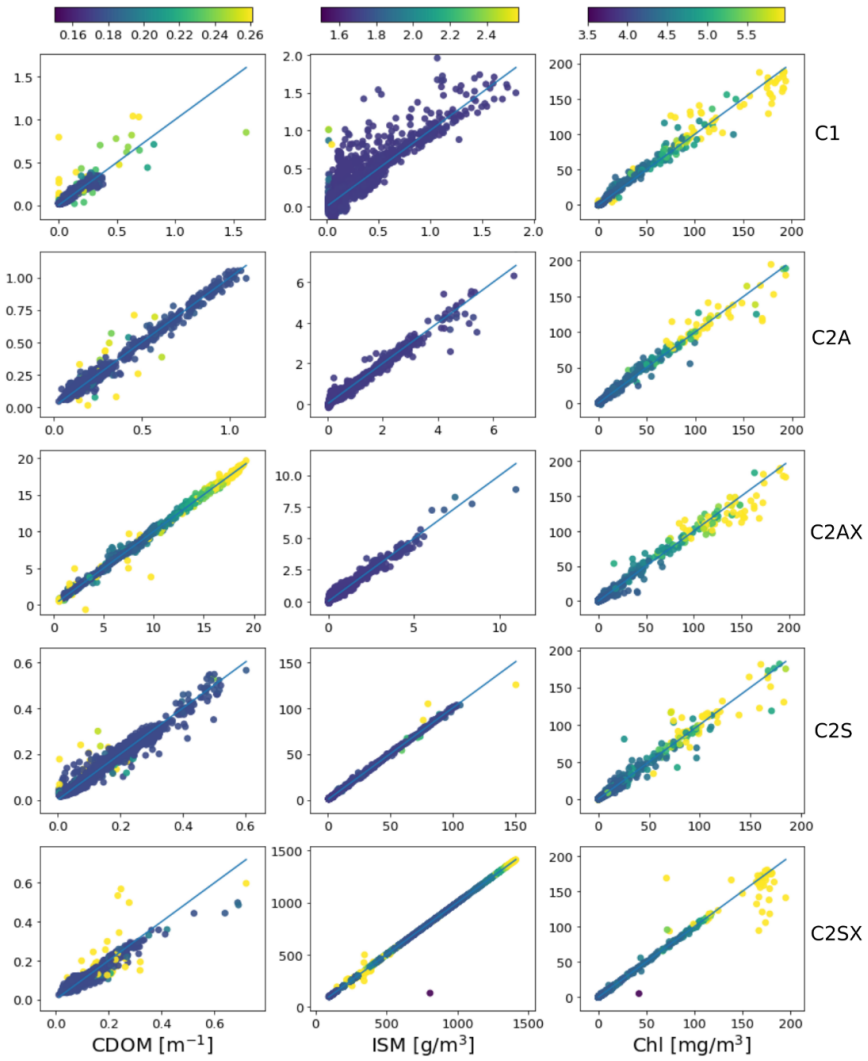


Fig. 7. Actual values (x-axis) versus values predicted by the DGP (y-axis) of test data, for each of the different water quality variables. The plots are divided by water type, and coloured according to predictive uncertainty: $2 * \sqrt{\sigma^2_{DGP}(\mathbf{x})}$. We see that the predictive uncertainty is generally quite conservative, however it does tend to flag the point of high prediction error with higher predictive uncertainty. This behaviour is preferable compared to the more erratic uncertainty estimates in previous attempts at modeling this data (Hieronymi et al., 2017).

types. The predictions and uncertainties are calculated as the weighted sum of the retrievals of all class-specific NNs. On the other hand, the DGP approach is a single model (pr. output) with three layers and 500 inducing points and 5 GPs per layer. This makes for a more elegant formulation in which there is less choice necessary with respect to model decisions (e.g. 3–4 layers was usually enough to fit the data), and which calculates all uncertainties simultaneously with the retrievals. The main success of the experiment is the increase in the accuracy of the Chl quantification for the five different water types. The errors decrease up to a factor 6 in the C2SX water type (see Table 3). Uncertainty estimation was found to be a hard problem as previously shown in (Hieronymi et al., 2017). This is likely due to the fact that the dataset exhibits high variability in regions of both low numeric values and high ones (orders of magnitude from 10^{-2} to 10^{2}). Nevertheless, the DGP shows some advantages over the ONNS approach: it flags many of the outliers with high predictive uncertainty, and provides more conservative uncertainty estimates than the ONNS which assigns high uncertainty to predictions with low errors as well as vice versa.

4. Conclusions

We introduced the use of deep GPs and the doubly stochastic variational inference procedure for remote sensing applications involving parameter retrieval and model inversion. The applied deep GP model can efficiently handle the biggest challenges nowadays: dealing with big data problems while being expressive enough to account for highly nonlinear problems. We successfully illustrated its performance in two scenarios involving optical simulated Sentinel-3 OLCI data and IASI sounding data, and for different data sizes, dimensionality, and distributions of the target bio-geo-physical parameters.

We showed how DGP benefits from its hierarchical structure and consistently outperforms both full and sparse GPs in all cases and situations on the data at hand. Depth plays a fundamental role but the main increase in performance is achieved when going from shallow to deep Gaussian Processes, i.e. going from 1 to 2 layers. Higher number of layers showed little improvement and a certain risk of overfitting because of model over-parameterization. Importantly, unlike a standard

GP, the DGP model is inherently sparse and scales linearly with the training set size.

We would like to stress that the used DGPs could make a difference in the two applications introduced here, now and in the near future. For instance, neural networks made a revolution in the last decade for the estimation of atmospheric variables from infrared sounding data (Blackwell, 2005; Blackwell et al., 2008). Later, in (Camps-Valls et al., 2012) we showed that kernel methods can outperform neural networks in these scenarios of high-input and output data dimensionality, but are more computationally costly and memory demanding when bigger datasets are available. With DGPs these shortcomings are remedied: they are more expressive and accurate than standard kernel ridge regression (i.e. one-layer plain GPs), computationally much more efficient, and additionally provide a principled way to derive confidence intervals for the predictions. The problem of estimating temperature and moisture variables was successfully addressed with DGPs, and results were more accurate both over land/ocean, and for different latitudes, climatic zones and biomes. Furthermore, the experimental results for prediction of CDOM, ISM and Chl-a showed that it was possible to make a 3-layer DGP outperform a Neural Network based algorithm proposed in the literature. Although uncertainty quantification is difficult, as seen in (Hieronymi et al., 2017), it is an advantage that training a DGP automatically yields uncertainty estimates, avoiding the need to train additional uncertainty neural networks.

The DGP model has demonstrated excellent capabilities in terms of accuracy and scalability, but certainly some future improvements are needed. It does not escape our attention that, as has been shown for deep convolutional neural networks, convolutional models can improve predictions when there is clear spatial structure (Malmgren-Hansen et al., 2017). Currently there are some efforts in the direction of convolutional GPs (Van der Wilk et al., 2017), but performance is still not comparable to a convolutional neural network (CNN). As shown here and in the literature, DGPs scale very well to large amounts of data, and have been trained on problems with 10^9 datapoints (Salimbeni and Deisenroth, 2017). As of now, however, feed forward neural networks are still generally faster to train, which is not surprising as the DGP is learning a predictive distribution instead of a single point estimate. Lastly, when it comes to dealing with missing data and mixed data modalities, random forest regression has often been found to be more flexible than other methods. There is interesting work however addressing the missing data problem for GPs (Damianou and Lawrence, 2015).

The more we incorporate machine learning in the pipeline when modeling physical systems, the more important uncertainty estimation and error propagation become. Encoding prior knowledge about input noise into a standard GP in a parameter retrieval setting, it has been shown that improved uncertainty estimation can be achieved (Johnson et al., 2019). The same approach can be imagined with a DGP model, which in the future could additionally improve its uncertainty estimates.

Declaration of Competing Interest

None.

Acknowledgements

This work is funded by the European Research Council (ERC) under the ERC-CoG-2014 SEDAL project (grant agreement 647423), the Spanish Ministry of Economy and Competitiveness through projects TIN2015-64210-R, DPI2016-77869-C2-2-R, and the Spanish Excellence Network TEC2016-81900-REDT. Pablo Morales-Álvarez is supported by *La Caixa* Banking Foundation (ID 100010434, Barcelona, Spain) through *La Caixa* Fellowship for Doctoral Studies LCF-BQ-ES17-11600011.

The authors want to thank J. Emmanuel Johnson and Dr. Valero Laparra (Universitat de València) for preparing the IASI data; and Martin Hieronymi from Helmholtz-Zentrum Geesthacht and the C2X project for the C2X data set. Hurricane Isabel data used in Fig. 1 is produced by the Weather Research and Forecast (WRF) model, courtesy of NCAR and the U.S. National Science Foundation (NSF). We give thanks to Dr. David Malmgren-Hansen (DTU, Denmark) for generating Fig. 4.

Appendix A. The Fully Independent Training Conditional (FITC) method

Specifically, FITC approximates the model by assuming: i) conditional independence between train and test latent variables $\mathbf{f}, \mathbf{f}_*$ given the inducing points $\mathbf{u}$; and ii) a factorized (fully independent) distribution for $\mathbf{f}$ given $\mathbf{u}$. Under these hypothesis, the approximated model for FITC (which replaces the exact $p(\mathbf{f}, \mathbf{f}_*)$) is:

$$\widetilde{p}(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q}_{ff} + \text{diag}(\mathbf{K}_{ff} - \mathbf{Q}_{ff}) & \mathbf{Q}_{f*} \\ \mathbf{Q}_{*f} & \mathbf{K}_{**} \end{bmatrix}\right), \tag{2}$$

where we abbreviate $\mathbf{Q}_{ab} = \mathbf{K}_{au}\mathbf{K}_{uu}^{-1}\mathbf{K}_{ub}$. With this approximation, the observation model $p(\mathbf{y}|\mathbf{f}, \rho^2)$ can be marginalized and the new matrix to be inverted is $(\mathbf{Q}_{ff} + \text{diag}(\mathbf{K}_{ff} - \mathbf{Q}_{ff}) + \sigma^2\mathbf{I})$. Interestingly, this new low-rank-plus-diagonal matrix can be inverted with $O(nm^2)$ cost by applying the Woodbury matrix identity (Rasmussen and Williams, 2006). Finally, the most common practice for the inducing locations $\mathbf{Z}$ is to estimate them along with the kernel hyperparameters and ρ^2 by maximizing the marginal likelihood (Snelson and Ghahramani, 2006).

Regarding the predictive distribution, FITC leverages the conditional independence of $\mathbf{f}_*$ from $\mathbf{f}$ given $\mathbf{u}$. Recall that the predictive distribution for a standard GP on a new $\mathbf{x}_*$ is a Gaussian with mean and covariance given by (Rasmussen and Williams, 2006):

$$\mu_{\text{GP}} = \mathbf{K}_{*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{y},$$
$$\sigma^2_{\text{GP}} = \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f}*}.$$

Consequently, the predictive mean and variance for FITC is (Snelson and Ghahramani, 2006):

$$\mu_{\text{FITC}} = \mathbf{Q}_{*\mathbf{f}}(\mathbf{Q}_{\mathbf{ff}} + \text{diag}(\mathbf{K}_{ff} - \mathbf{Q}_{ff}) + \sigma^2\mathbf{I})^{-1}\mathbf{y},$$
$$\sigma^2_{\text{FITC}} = \mathbf{K}_{**} - \mathbf{Q}_{*\mathbf{f}}(\mathbf{Q}_{\mathbf{ff}} + \text{diag}(\mathbf{K}_{ff} - \mathbf{Q}_{ff}) + \sigma^2\mathbf{I})^{-1}\mathbf{Q}_{\mathbf{f}*}.$$

Appendix B. Doubly Stochastic Variational Inference for DGP

The approach followed in (Salimbeni and Deisenroth, 2017) to do inference in DGPs relies on variational inference (VI). The general idea of VI is to transform the problem of posterior distribution computation into an optimization one, by introducing a parametric family of candidate posterior distributions. Moreover, in VI this optimization is solved together with the maximization of the marginal log-likelihood $\log p(\mathbf{y})$. More specifically, since the selected family will not usually contain the exact posterior, the target of the optimization will be a lower bound on $\log p(\mathbf{y})$. This is the so-called Evidence Lower Bound (ELBO) (Blei et al., 2017).

The proposed family of posterior distributions in (Salimbeni and Deisenroth, 2017) is

$$q(\{\mathbf{f}^l, \mathbf{u}^l\}|\{\mathbf{z}^l, \mathbf{m}^l, \mathbf{S}^l\}) = \prod_{l=1}^{L} p(\mathbf{f}^l|\mathbf{u}^l;\mathbf{f}^{l-1}, \mathbf{z}^{l-1}) q(\mathbf{u}^l). \tag{3}$$

Notice that the first factor is the prior conditional of Eq. (1), and keeps correlations between layers. The second is taken Gaussian with mean $\mathbf{m}^l$ and full covariance $\mathbf{S}^l$ (which are variational parameters of the parametric family, to be estimated). With this posterior, the ELBO for the marginal log-likelihood $\log p(\mathbf{y})$ is then obtained[3]:

$$\log p(\mathbf{y}) = \log \int \frac{q(\{\mathbf{f}^l, \mathbf{u}^l\})}{q(\{\mathbf{f}^l, \mathbf{u}^l\})} p(\mathbf{y}, \{\mathbf{f}^l, \mathbf{u}^l\}) d\mathbf{f}^l d\mathbf{u}^l \geqslant \sum_{i=1}^{n} \mathbb{E}_{q(f_i^L)}[\log p(y_i|f_i^L)] - \sum_{l=1}^{L} \mathrm{KL}(q(\mathbf{u}^l)||p(\mathbf{u}^l;\mathbf{z}^{l-1})). \tag{4}$$

Observe that the second term is tractable, as the KL divergence between Gaussians can be computed in closed form (Rasmussen and Williams, 2006). However, the expectation in the first term involves the marginals of the posterior at the last layer, $q(f_i^L)$. Next we see that, whereas this distribution is analytically intractable, it can be sampled efficiently using univariate Gaussians.

Indeed, marginalizing out the inducing points in Eq. (3), the posterior for the GP layers $\{\mathbf{f}^l\}$ is

$$q(\{\mathbf{f}^l\}) = \prod_{l=1}^{L} q(\mathbf{f}^l|\mathbf{m}^l, \mathbf{S}^l;\mathbf{f}^{l-1}, \mathbf{z}^{l-1}) = \prod_{l=1}^{L} \mathcal{N}(\mathbf{f}^l|\widetilde{\boldsymbol{\mu}}^l, \widetilde{\boldsymbol{\Sigma}}^l), \tag{5}$$

where the vector $\widetilde{\boldsymbol{\mu}}^l$ is given by $[\widetilde{\boldsymbol{\mu}}^l]_i = \mu_{\mathbf{m}^l,\mathbf{z}^{l-1}}(f_i^{l-1})$ and the $n \times n$ matrix $\widetilde{\boldsymbol{\Sigma}}^l$ by $[\widetilde{\boldsymbol{\Sigma}}^l]_{ij} = \Sigma_{\mathbf{S}^l,\mathbf{z}^{l-1}}(f_i^{l-1}, f_j^{l-1})$. The explicit expression for the functions $\mu_{\mathbf{m}^l,\mathbf{z}^{l-1}}$ and $\Sigma_{\mathbf{S}^l,\mathbf{z}^{l-1}}$ can be found in (Salimbeni and Deisenroth, 2017, Eqs. (7-8)). The key point here is to observe that, although the distribution in Eq. (5) is fully coupled between layers (and thus the posterior in the last layer is analytically intractable), the i-th marginal at each layer $\mathcal{N}(f_i^l|[\widetilde{\boldsymbol{\mu}}^l]_i, [\widetilde{\boldsymbol{\Sigma}}^l]_{ii})$ only depends on the corresponding i-th input of the previous layer. This allows one to recursively sample $\widehat{f}_i^1 \to \widehat{f}_i^2 \to \ldots \to \widehat{f}_i^L$ from all the layers up to the last one by means of just univariate Gaussians. Specifically, $\varepsilon_i^l \sim \mathcal{N}(0, 1)$ is first sampled and then for $l = 1, \ldots, L$:

$$\widehat{f}_i^l = \mu_{\mathbf{m}^l,\mathbf{z}^{l-1}}(\widehat{f}_i^{l-1}) + \varepsilon_i^l \cdot \sqrt{\Sigma_{\mathbf{S}^l,\mathbf{z}^{l-1}}(\widehat{f}_i^{l-1}, \widehat{f}_i^{l-1})}. \tag{6}$$

Now, the expectation in the ELBO (recall Eq. (4)) can be approximated with a Monte Carlo sample generated with Eq. (6). This provides the first source of stochasticity. Since the ELBO factorizes across data points and the samples can be drawn independently for each point i, scalability is achieved through sub-sampling the data in mini-batches. This is the second source of stochasticity, which motivates the naming of this *doubly stochastic* inference scheme.

The ELBO is maximized with respect to the variational parameters $\mathbf{m}^l$, $\mathbf{S}^l$, the inducing locations $\mathbf{z}^l$, and the kernel and likelihood hyperparameters ϑ^l, ρ^2 (which, to alleviate the notation, have not been included in the equations). Notice that the complexity to evaluate the ELBO and its gradients is $O(n_b m^2 (D^1 + \ldots + D^L))$, where n_b is the size of the mini-batch used, and D_l is the number of hidden units in each layer (which were set to one in this section). As mentioned before, this extends the scalability of the (shallow) sparse GP approximation SVGP (Hensman et al., 2013) to hierarchical models, including the batching capacity.

Appendix C. Predictions

To predict in a new $\mathbf{x}_*$ in DGPs, Eq. (6) is used to sample S times[4] from the posterior up to the $(L-1)$-th layer using the test location as initial input. This yields a set $\{f_*^{L-1}(s)\}_{s=1}^S$ with S samples. Then, the density over f_*^L is given by the Gaussian mixture (recall that all the terms in Eq. (5) are Gaussians):

$$q(f_*^L) = \frac{1}{S} \sum_{s=1}^{S} q(f_*^L|\mathbf{m}^L, \mathbf{S}^L;\mathbf{f}_*^{L-1}(s), \mathbf{z}^{L-1}).$$

Appendix D. Climate zones and biome classification

The climate zones data were taken from the Köppen-Geiger climate classification maps.[5] The biome zones are aggregations of several classes from the standard International Geosphere-Biosphere Programme (IGBP) biome classification.[6] The tables below show the IGBP class names and the aggregations performed in this work which are used in Table 2.

[3] The key idea here is that the prior conditionals of Eq. (3) cancel with those of Eq. (1). This makes Eq. (3) a very convenient posterior choice.

[4] This S is related to the first source of stochasticity and, theoretically, the higher the better. In practice, results become stable after a few samples. Here, S was set to 200.

[5] See http://koeppen-geiger.vu-wien.ac.at/.

[6] For an implementation of the IGBP biome map with 0.05 degree spatial resolution see https://lpdaac.usgs.gov/products/mcd12c1v006/.

Table 4
IGBP biome class names as well as the aggregations of the IGBP classes performed in this work and their corresponding names used in Table 2.

IGBP name	Acronym
Water	WATWAT
Evergreen Needleleaf forest	ENF
Evergreen Broadleaf forest	EBF
Deciduous Needleleaf forest	DNF
Deciduous Broadleaf forest	DBF
Mixed forest	MF
Closed shrublands	CSH
Open shrublands	OSH
Woody savannas	WSA
Savannas	SAV
Grasslands	GRA
Permanent wetlands	WET
Croplands	CRO
Urban and built-up	URB
Cropland/Natural vegetation mosaic	CVM
Snow and ice	SNO
Barren or sparsely vegetated	BSV
Aggregate name	**Aggregated classes**
Needle-leaf Forest	ENF + DNF
Evergreen Broadleaf Forest	EBF
Decidious Broadleaf Forest	DBF
Mixed forest	MF
Shrublands	CSH + OSH
Savannas	WSA + SAV
Herbaceous	GRA
Cultivated	CRO

Appendix E. Ocean color results in logscale

We include here the results of Fig. 7 in log-scale in order to highlight the behaviour of the model when predicting on low numerical values of the parameters.

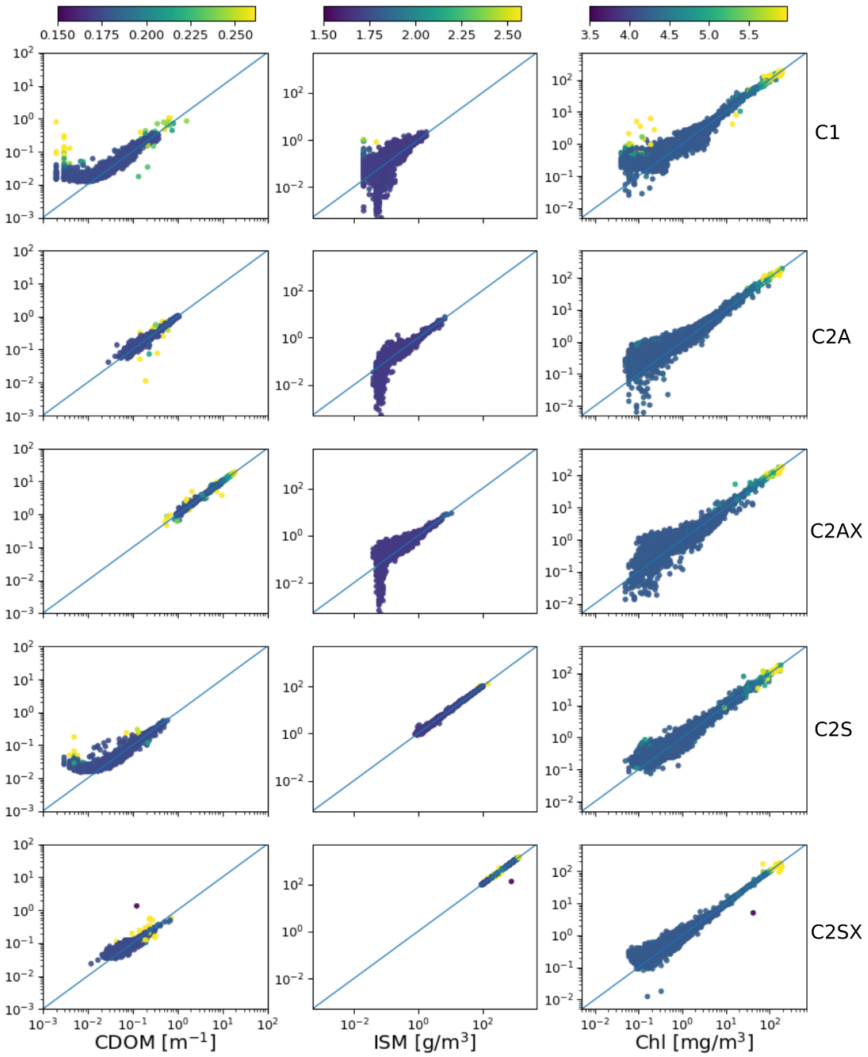


Fig. 8. Actual values (x-axis) versus values predicted by the DGP (y-axis) of test data in log scale, for each of the different water quality variables. The plots are divided by water type, and coloured according to predictive uncertainty: $2 * \sqrt{\sigma^2_{DGP}(\mathbf{x})}$.

References

Aires, F., 2002. A regularized neural net approach for retrieval of atmospheric and surface temperatures with the IASI instrument. J. Appl. Meteorol. 41, 144–159.

Bauer, M., van der Wilk, M., Rasmussen, C., 2016. Understanding probabilistic sparse Gaussian process approximations. In: Advances in Neural Information Processing Systems, 2016, pp. 1533–1541.

Bayesian methodology for inverting satellite ocean-color data. Remote Sens. Environ., vol. 159, 2015, pp. 332 – 360.

Blackwell, W.J., 2005. A neural-network technique for the retrieval of atmospheric temperature and moisture profiles from high spectral resolution sounding data. IEEE Trans. Geosc. Rem. Sens. 43 (11), 2535–2546.

Blackwell, W. J., Pieper, M., Jairam, L., 2008. Neural network estimation of atmospheric profiles using AIRS/IASI/AMSU data in the presence of clouds. In: Suzuki, A.M.L.M.J. L.M. (Ed.), Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques, and Applications II, Proceedings of SPIE Vol. 7149, Bellingham, WA, 2008.

Blei, D.M., Kucukelbir, A., McAuliffe, J.D., 2017. Variational inference: a review for statisticians. J. Am. Stat. Assoc. 112 (518), 859–877.

Bukata, R., Jerome, J., Kondratyev, K., Pozdnaykov, D., 1995. Optical properties and remote sensing of Inland and Coastal Waters. CRC Press, Boca Raton, FL.

Camps-Valls, G., Bruzzone, L. (Eds.), 2009. Kernel methods for Remote Sensing Data Analysis. Wiley & Sons, UK.

Camps-Valls, G., Muñoz-Marí, J., Gómez-Chova, L., Guanter, L., Calbet, X., 2012. Nonlinear statistical retrieval of atmospheric profiles from MetOp-IASI and MTG-IRS infrared sounding data. IEEE Trans. Geosci. Remote Sens. 50 (5 Part 2), 1759–1769.

Camps-Valls, G., Munoz-Mari, J., Gomez-Chova, L., Guanter, L., Calbet, X., 2012. Nonlinear statistical retrieval of atmospheric profiles from MetOp-IASI and MTG-IRS infrared sounding data. IEEE Trans. Geosci. Remote Sens. 50 (5), 1759–1769.

Camps-Valls, G., Svendsen, D., Martino, L., Munoz-Mari, J., Laparra, V., Campos-Taberner, M., Luengo, D., 2018. Physics-aware gaussian processes in remote sensing. Appl. Soft Comput. 68, 69–82.

Damianou, A., 2015. Deep Gaussian processes and variational propagation of uncertainty. Ph.D. thesis. University of Sheffield.

Damianou, A., Lawrence, N., 2013. Deep Gaussian processes, in: Artificial Intelligence

and Statistics, 2013, pp. 207–215.
Damianou, A., Lawrence, N.D., 2015. Semi-described and semi-supervised learning with gaussian processes, arXiv preprint arXiv:1509.01168.
Dekker, A., Peters, S., Vos, R., Rijkeboer, M., 2001. Remote sensing for inland water quality detection and monitoring: State-of-the-art application in Friesland waters. In: van Dijk, A., Bos, M.G. (Eds.) GIS and Remote Sensing Techniques in Land- and Water-management, Springer, 2001.
Doerffer, R., Schiller, H., 2007. The MERIS Case 2 water algorithm. Int. J. Remote Sens. 28 (3–4), 517–535.
EUMETSAT, IASI Level 1: Product Guide, EUM/OPS-EPS/MAN/04/0032, 2014.
Furfaro, R., Morris, R.D., Kottas, A., Taddy, M., Ganapol, B.D., 2006. A Gaussian Process Approach to Quantifying the Uncertainty of Vegetation Parameters from Remote Sensing Observations, AGU Fall Meeting Abstracts (2006) A261+.
Camps-Valls, G., Tuia, D., Gómez-Chova, L., Malo, J. (Eds.), 2011. Remote Sensing Image Processing, Morgan & Claypool, 2011.
Gustau Camps-Valls, J.R.M.R., Sejdinovic Dino, 2019. A perspective on gaussian processes for earth observation. Natl. Sci. Rev.
Hensman, J., Fusi, N., Lawrence, N.D., 2013. Gaussian processes for big data. In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence. AUAI Press, pp. 282–290.
Hieronymi, M., Kraseman, H., Ruescas, A., Brockmann, C., Steinmetz, F., Tilstone, G., Simis, S., 2015. Algorithm theoretical basis document, Tech. rep., Case 2 eXtreme Project, ESA (2015).
Hieronymi, M., Krasemann, H., Mueller, D., Brockmann, C., Ruescas, A., Stelzer, K., Nechad, B., Ruddick, K., Simis, S., Tisltone, G., Steinmetz, F., Regner, P., 2016. Ocean colour remote sensing of extreme Case-2 waters. In: Proceedings of the LPS, Living Planet Symposium.
Hieronymi, M., Mueller, D., Doerffer, R., 2017. The OLCI Neural Network Swarm (ONNS): A bio-geo-optical algorithm for open ocean and coastal waters. Front. Marine Sci. 4, 140.
Huang, H.L., Smith, W.L., Woolf, H.M., 1992. Vertical resolution and accuracy of atmospheric infrared sounding spectrometers. J. Appl. Meteor. 31, 265–274.
Johnson, J.E., Laparra, V., Camps-Valls, G., 2019. Accounting for input noise in gaussian process parameter retrieval. IEEE Geosci. Remote Sens. Lett.
Jung, M., Reichstein, M., Schwalm, C.R., Huntingford, C., Sitch, S., Ahlström, A., Arneth, A., Camps-Valls, G., Ciais, P., Friedlingstein, P., et al., 2017. Compensatory water effects link yearly global land co 2 sink changes to temperature. Nature 541 (7638), 516–520.
Jung, M., Reichstein, M., Schwalm, C.R., Huntingford, C., Sitch, S., Ahlström, A., Arneth, A., Camps-Valls, G., Ciais, P., Friedlingstein, P., Gans, F., Ichii, K., Jain, A.K., Kato, E., Papale, D., Poulter, B., Raduly, B., Rödenbeck, C., Tramontana, G., Viovy, N., Wang, Y.-P., Weber, U., Zaehle, S., Zeng, N., 2017. Compensatory water effects link yearly global land co2 sink changes to temperature. Nature 541 (7638), 516–520.
Kallio, K., 2006. Optical properties of Finnish lakes estimated with simple bio-optical models and water quality monitoring data. Hydrol. Res. 37 (2), 183–204.
Kraseman, H., Hieronymi, M., Simis, S., Steinmetz, F., Tilstone, G., Nechad, B., Kraemer, U., 2016. Database for task 2, technical note, Tech. rep., Case 2 eXtreme Project, ESA (2016).
Laparra, V., Malo, J., Camps-Valls, G., 2015. Dimensionality reduction via regression in hyperspectral imagery. IEEE J. Select. Top. Signal Process. 9 (6), 1026–1036.
Laparra, V., Muñoz-Marí, J., Gómez-Chova, L., Calbet, X., Camps-Valls, G., 2017. Nonlinear statistical retrieval of surface emissivity from iasi data. In: IEEE International and Remote Sensing Symposium (IGARSS), 2017.
Liang, S., 2008. Advances in Land Remote Sensing: System, Modeling, Inversion and Applications. Springer Verlag, Germany.
Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: a meta-analysis and review. ISPRS J. Photogram. Remote Sens. 152, 166–177.
Malmgren-Hansen, D., Laparra, V., Nielsen, A.A., Camps-Valls, G., 2017. Spatial noise-aware temperature retrieval from infrared sounder data. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2017, pp. 17–20.
Malmgren-Hansen, D., Laparra, V., Nielsen, A.A., Camps-Valls, G., 2019. Statistical retrieval of atmospheric profiles with deep convolutional neural networks. ISPRS J. Photogram. Remote Sens. 158, 231–240.
Mobley, C., Sundman, L.K., 2013. Hydrolight 5.2, ecolight 5.2, technical documentation, Tech. rep., Sequoia Sci., Inc., Mercer Island, Wash, 2013.
Morales-Alvarez, P., Pérez-Suay, A., Molina, R., Camps-Valls, G., 2017. Remote sensing image classification with large-scale gaussian processes. IEEE Trans. Geosci. Remote Sens. 56 (2), 1103–1114.
Morales-Álvarez, P., Pérez-Suay, A., Molina, R., Camps-Valls, G., 2017. Remote sensing image classification with large-scale Gaussian processes. IEEE Trans. Geosci. Remote Sens. PP (99), 1–12.
Morel, M., Prieur, L., 1977. Analysis of variations in ocean color. Limnol. Oceanogr. 22 (4), 709–722.
Neal, R.M., 1996. Priors for infinite networks, in: Bayesian Learning for Neural Networks, Springer, 1996, pp. 29–53.
Prieur, L., Sathyendranath, S., 2019. An optical classification of coastal and oceanic waters based on the specific spectral absorption curves of phytoplankton pigments, dissolved organic matter, and other particulate materials. Limnol. Oceanogr., vol. 26.
Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. The MIT Press, New York.
Rivera-Caicedo, J.P., Verrelst, J., Muñoz-Marí, J., Camps-Valls, G., Moreno, J., 2017. Hyperspectral dimensionality reduction for biophysical variable statistical retrieval. ISPRS J. Photogram. Remote Sens. 132, 88–101.
Rodgers, C.D., 2000. Inverse Methods for Atmospheric Sounding: Theory and Practice. World Scientific Publishing Co., Ltd.
Rojo-Álvarez, J., Martínez-Ramón, M., Muñoz Marí, J., Camps-Valls, G., 2018. Digital Signal Processing with Kernel Methods. Wiley & Sons, UK.
Ruescas, A.B., Hieronymi, M., Mateo-Garcia, G., Koponen, S., Kallio, K., Camps-Valls, G., 2018a Machine learning regression approaches for colored dissolved organic matter (cdom) retrieval with s2-msi and s3-olci simulated data. Remote Sens., vol. 10, 5.
Ruescas, A., Mateo-Garcia, G., Hieronymi, M., Camps-Valls, G., 2018b. Retrieval of case 2 water quality parameters with machine learning. In: Proceedings of the IGARSS 2018, IGARSS, 2018.
Ruescas, A.B., Hieronymi, M., Mateo-Garcia, G., Koponen, S., Kallio, K., Camps-Valls, G., 2018c. Machine learning regression approaches for colored dissolved organic matter (cdom) retrieval with s2-msi and s3-olci simulated data. Remote Sens., vol 10, 5.
Salimbeni, H., Deisenroth, M., 2017. Doubly stochastic variational inference for deep Gaussian processes. In: Advances in Neural Information Processing Systems, 2017, pp. 4591–4602.
Sarkar, D., Osborne, M.A., Adcock, T.A.A., 2019. Spatiotemporal prediction of tidal currents using gaussian processes. J. Geophys. Res.: Oceans, vol. 124, 4, pp. 2697–2715.
Schneider, S., Murphy, R.J., Melkumyan, A., 2014. Evaluating the performance of a new classifier–the gp-oad: A comparison with existing methods for classifying rock type and mineralogy from hyperspectral imagery. ISPRS J. Photogram. Remote Sens. 98, 145–156.
Shi, J., Khan, M.E., Zhu, J., 2019. Scalable training of inference networks for gaussian-process models. In: International Conference on Machine Learning, pp. 5758–5768.
Siméoni, D., Singer, C., Chalon, G., 1997. Infrared atmospheric sounding interferometer. Acta Astronaut. 40, 113–118.
Snelson, E., Ghahramani, Z., 2006. Sparse Gaussian processes using pseudo-inputs. Adv. Neural Inform. Process. Syst. 1257–1264.
Svendsen, D.H., Morales-Álvarez, P., Molina, R., Camps-Valls, G., 2018. Deep gaussian processes for geophysical parameter retrieval. In: IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2018, pp. 6175–6178.
Titsias, M.K., Lawrence, N.D., 2010. Bayesian Gaussian process latent variable model. In: International Conference on Artificial Intelligence and Statistics, pp. 844–851.
Tournier, B., Blumstein, D., Cayla, F., Chalon, G., 2002. IASI level 0 and 1 processing algorithms description. In: Proc. of ISTCXII Conference, 2002.
Tramontana, G., Jung, M., Camps-Valls, G., Ichii, K., Raduly, B., Reichstein, M., Schwalm, C.R., Arain, M.A., Cescatti, A., Kiely, G., Merbold, L., Serrano-Ortiz, P., Sickert, S., Wolf, S., Papale, D., 2016. Predicting carbon dioxide and energy fluxes across global fluxnet sites with regression algorithms. Biogeosci. Discuss. 2016, 1–33.
Van der Wilk, M., Rasmussen, C.E., Hensman, J., 2017. Convolutional gaussian processes. In: Advances in Neural Information Processing Systems, 2017, pp. 2849–2858.
Verrelst, J., Alonso, L., Camps-Valls, G., Delegido, J., Moreno, J., 2012. Retrieval of vegetation biophysical parameters using gaussian process techniques. IEEE Trans. Geosc. Rem. Sens. 50 (5 PART 2), 1832–1843.
Verrelst, J., Rivera, J., Moreno, J., Camps-Valls, G., 2013. Gaussian processes uncertainty estimates in experimental Sentinel-2 LAI and leaf chlorophyll content retrieval. ISPRS J. Photogram. Remote Sens. 86, 157–167.
Verrelst, J., Rivera, J.P., Veroustraete, F., Muñoz-Marí, J., Clevers, J.G., Camps-Valls, G., Moreno, J., 2015. Experimental sentinel-2 lai estimation using parametric, non-parametric and physical retrieval methods–a comparison. ISPRS J. Photogram. Remote Sens. 108, 260–272.
Verrelst, J., Rivera, J.P., Gitelson, A., Delegido, J., Moreno, J., Camps-Valls, G., 2016. Spectral band selection for vegetation properties retrieval using gaussian processes regression. Int. J. Appl. Earth Observ. Geoinform. 52, 554–567.
Wernecke, A., Edwards, T.L., Nias, I.J., Holden, P.B., Edwards, N.R., 2019. Spatial probabilistic calibration of a high-resolution amundsen sea embayment ice-sheet model with satellite altimeter data. The Cryosphere Discussions 2019, 1–21.

Joint Gaussian Processes for Biophysical Parameter Retrieval

Daniel Heestermans Svendsen, Luca Martino, Manuel Campos-Taberner, Francisco Javier García-Haro, and Gustau Camps-Valls, *Senior Member, IEEE*

***Abstract*— Solving inverse problems is central in geosciences and remote sensing. The radiative transfer models (RTMs) represent mathematically the physical laws that rule the phenomena in remote sensing applications (forward models). The numerical inversion of the RTM equations is a challenging and computationally demanding problem. For this reason, often the application of a simpler statistical regression is preferred. In general, the regression models predict the biophysical parameter of interest from the corresponding received radiance, learning a mapping from *in situ* data. However, this approach does not employ the physical information encoded in the RTMs. An alternative strategy, which attempts to include the physical knowledge, consists in learning a regression model trained using simulated data by an RTM code. In this paper, we introduce a nonlinear nonparametric regression model that combines the benefits of the two aforementioned approaches. The inversion is performed considering jointly both real observations and RTM-simulated data. The proposed joint Gaussian process (JGP) provides a solid framework for exploiting the regularities between the two types of data, in order to perform inverse modeling. The JGP automatically detects the relative quality of the simulated and real data, and combines them properly. This occurs by learning an additional hyperparameter with respect to a standard Gaussian process model, so that the novel scheme is at the same time simple and robust, i.e., capable of adapting to different scenarios. The advantages of the JGP method compared with benchmark strategies are shown considering synthetic and real data in different experiments. Specifically, we consider leaf area index retrieval from Landsat data combined with simulated data generated by the PROSAIL model.**

***Index Terms*— Gaussian process (GP) regression, inverse modeling, kernel methods, multitask learning, PROSAIL, radiative transfer model (RTM), vegetation monitoring.**

I. Introduction

SOLVING *forward* and *inverse* problems lies at the heart of research in geoscience, remote sensing, and physics in general. The *forward modeling* problem consists mainly in determining the physical laws that govern complex phenomena (e.g., modeling the response of a sensor to different physical inputs). Then, the designed model is implemented and tested in different scenarios in order to study the ability to explain the observed physical phenomena. These forward mechanistic models are also used to generate artificial measurements [1]. In this paper, we focus on radiative transfer models (RTMs), which play the role of forward models in remote sensing applications of biophysical parameter estimation.

The aim of the *inverse* problem is to determine the underlying physical conditions that correspond to a given set of real obtained measurements. That is, it attempts to make an inference about the physical parameters from sensory data. A very relevant problem is that of estimating vegetation properties from remote sensing observations. Accurate inverse models help determine the phenological stage and health status (e.g., development, productivity, and stress) of crops and forests [2], which has important societal, environmental, and economical implications, given the evergrowing demand for biofuel and food. Leaf chlorophyll content (Chl), leaf area index (LAI), biomass, and fractional vegetation cover are among the most important vegetation parameters [3], [4].

RTMs are typically used to implement the forward direction [5], [6]. However, inverting RTMs directly is very complex, because the number of unknowns is generally larger than the number of independent radiometric information [7]. In addition, estimating physical parameters from RTMs is hampered by the presence of high levels of uncertainty and noise, primarily associated with atmospheric conditions and sensor calibration, sun angle, and viewing geometry, as well as the poor sampling of the parameter space in most of the applications. This translates into inverse problems where spectra deemed similar may correspond to diverse solutions. This gives rise to nondetermination and ill-posed problems.

Methods for solving the inverse problems (i.e., parameter retrieval) can be classified in three main families: *statistical*, *physical* (also known as *numerical*), and *hybrid* inversion methods [8]. The *statistical inversion* approach consists in applying a regression method in order to predict a biogeophysical parameter of interest such as LAI, given observations obtained by the satellite. The regression models are trained using a collected data set with data pairs formed by measurements obtained by the satellite (as input, e.g., reflectances) and from the corresponding parameter of interest (as output, e.g., LAI) measured *in situ*. Note that this approach

Manuscript received June 16, 2017; revised October 11, 2017; accepted October 21, 2017. This work was supported by the European Research Council (ERC) through the ERC-CoG-2014 SEDAL Project under Grant 647423. *(Corresponding author: Daniel Heestermans Svendsen.)*

D. H. Svendsen, L. Martino, and G. Camps-Valls are with the Image Processing Laboratory, Universitat de València, 46980 València, Spain (e-mail: daniel.svendsen@uv.es).

M. Campos-Taberner and F. J. García-Haro are with the Departament de Física de la Terra i Termodinàmica, Facultat de Física, Universitat de València, 46010 València, Spain.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2767205

is purely statistical, since no physical information is employed for parameter retrieval.

Perhaps the most widely used approach in remote sensing is the *physical or numerical inversion*, which uses the information provided by the physical laws in parameter retrieval. Given a measured spectrum (e.g., from a satellite) and a suitable forward model, the idea is to compare with RTM-generated spectra in order to find the corresponding parameter of interest. This intuitive approach for the inversion of RTMs is based on searching for similar spectra in a lookup table, based on some similarity measure, and assigning the closest parameter [8], [9]. Alternatively, more sophisticated strategies have been considered, for instance, the use of computational algorithms based on Bayesian schemes [8], [10]. Compared with statistical inversion, the physical inversion is more computationally costly in general, but, it can yield more physically meaningful predictions for the parameters of interest. RTMs vary in complexity, based on the simplifications and assumptions made about the underlying physical phenomena. The cost of a more sophisticated physical model is computational complexity.

Finally, the last approach combines both the statistical and physical inversion. The *hybrid inversion* applies a statistical regression model learning from the artificial data generated by RTM simulations. That is, the hybrid inversion is similar to the statistical inversion but using simulated data only, instead of real data, for training the regression model [9], [11]–[14]. The rationale behind this approach is clear: exploit the flexibility and speed of statistical learning algorithms trained on physically meaningful data generated by an RTM. The advantage with respect to the statistical inversion is that larger data sets can be used for training (instead of a few, possibly not representative, real *in situ* measurements) and the physical knowledge encoded in the RTMs is indirectly employed. However, the quality of the inversion again depends dramatically on the quality of the artificial data generated, i.e., the ability of the RTMs to mimic real data in different scenarios.

Hybrid inversion is very powerful and practical when no *in situ* data are available. Indeed, hybrid inversion is currently an active field [11], [15], and is replacing physical inversion in many real applications and processing chains at local and global scale [16], [17]. However, it seems intuitive to let predictions be guided by actual measurements whenever they are present. The aim of this paper is to combine the statistical and hybrid inversions keeping the benefits of both the approaches. One trivial possible solution consists in training the regression model considering a single data set composed of the real and artificial data. However, when only a very few real *in situ* measurements are available, the method can be very sensitive to the incorporation of simulated data from RTMs. The reason being that this naive approach does not consider the differences between the statistical properties of the two types of data, and learns from both data sources without distinguishing them. As a consequence, the performance can be really poor and especially biased, depending on the quality of the RTM-simulated data. Another more sophisticated strategy consists in combining two different predictions obtained by independent regression models dedicated to each particular data set (or piece of information), thus performing a sort of model combination [18]–[21], [22, Ch. 8]. However, in this approach, the different data sets are analyzed separately; hence, the two regression models do not process all the available information, and may eventually lead to inconsistent (contradictory) predictions.

In this paper, we extend the hybrid inversion framework, proposing a statistical method, which performs nonlinear and nonparametric inversion blending both the real and simulated data with a suitable statistical approach. Our statistical model for parameter estimation is a Bayesian nonparametric approach known as Gaussian processes (GPs) [22]. GPs have yielded convincing results in recent years in many remote sensing and geoscience problems [23]–[25]. GPs provide the state-of-the-art prediction accuracy results and confidence intervals for the predictions, and allow model specification and interpretation in solid probabilistic terms (for an up-to-date review of GPs in remote sensing, see [14]). The proposed method in this paper, called a joint Gaussian process (JGP) exploits the information contained in both the data sets, and provides a solid framework for incorporating physical knowledge in GPs. It is particularly useful when the amount of *in situ* data is scarce and the simulated data are able to "fill in the gaps" of the input space, which incidentally is often the case in terrestrial campaigns. The JGP model is capable of automatically discovering the quality (noise, uncertainty) of each data set, and including this information in the regression model to balance their trustworthiness.

The remainder of this paper is organized as follows. Section II fixes notation, briefly reviews the GP framework, and introduces the JGP. We illustrate the performance of the JGP in a simple toy example, and comment on predictive mean in problems with multisource data sets. The JGP exploits the regularities between them, and provides a solid framework for incorporating physical knowledge in GP regression. Section III describes thoroughly the data used in the experiments. We rely on the retrieval of LAI from Landsat observations and PROSAIL simulated data. Both the real *in situ* measurements and the simulations were targeted to rice crop monitoring in three top-producing areas in Europe, but the scheme and model are general enough to be extended to other cases. We give empirical evidence of performance in Section IV. We performed exhaustive experiments and comparisons in terms of accuracy and robustness, and discuss on the elusive concepts of hyperparameter tuning and extrapolation when uneven uncertainty levels and data scarcity are involved. We conclude in Section V with some remarks and an outline of future work.

II. Joint Gaussian Processes

Model inversion through regression is an old, largely studied problem in statistics and machine learning, as well as in remote sensing and geosciences. A large class of regression models are available in the literature, such as random forests, neural networks, and kernel machines [8], [26], [27]. However, in the last decade, GPs have emerged as a solid framework to

tackle prediction problems in general and in remote sensing in particular [11], [14], [25], [28]. In this section, we first fix the notation, review the theory behind GPs, and propose the joint GP model.

A. Gaussian Process Regression

Let us consider a data set of n pairs of measurements, $\mathcal{D}_n := \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. The input data pairs used to fit the inverse machine learning model $f(\cdot)$ might come from either *in situ* field campaign data (statistical approach) or simulations by means of an RTM (hybrid approach). Either way, let us assume a model of the form

$$y_i = f(\mathbf{x}_i) + e_i, \quad e_i \sim \mathcal{N}\big(0, \sigma_e^2\big) \tag{1}$$

where $f(\mathbf{x})$ is an unknown latent function, $\mathbf{x} \in \mathbb{R}^d$, and σ_e^2 is the noise variance. Now, if we define the vectors $\mathbf{y} = [y_1, \ldots, y_n]^T$ and $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^T$, the conditional distribution of $\mathbf{y}$ given $\mathbf{f}$ becomes $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_e^2\mathbf{I})$, where $\mathbf{I}$ is the $n \times n$ identity matrix. At the heart of the GP approach is the assumption that $\mathbf{f}$ follows an n-dimensional Gaussian distribution, in this case with zero-mean $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$. The covariance matrix $\mathbf{K}$, which defines the GP, is determined by a kernel function $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, encoding the similarity between the input points [22]. The intuition here is the following: the more similar the inputs i and j are, according to some metric, the more correlated the output values i and j ought to be. The most common kernel function to account for such a similarity between points is the *squared exponential* (SE) $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$, which has some advantages: it is a universal kernel function, contains only one parameter that controls smoothness, and works in many diverse areas of application.

It can be easily verified that the marginal distribution of $\mathbf{y}$ can be written as

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{0}, \boldsymbol{C}_n)$$

where $\boldsymbol{C}_n = \mathbf{K} + \sigma_e^2\mathbf{I}$. Now, what we are really interested in is regression, that is, in predicting a new output value y_* given an input $\mathbf{x}_*$. The GP framework handles this by constructing a joint distribution over the training and test points

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{C}_n & \mathbf{k}_*^T \\ \mathbf{k}_* & c_* \end{bmatrix}\right)$$

where $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_n)]^T$ is an $n \times 1$ vector and $c_* = k(\mathbf{x}_*, \mathbf{x}_*) + \sigma_e^2$. Using standard manipulation of joint normally distributed variables [29], we can arrive at a distribution over y_* conditioned on the training data. This is a normal distribution with predictive mean and variance given by

$$\mu_{\mathrm{GP}}(\mathbf{x}_*) = \mathbf{k}_*^T(\mathbf{K} + \sigma_e^2\mathbf{I}_n)^{-1}\mathbf{y}, \tag{2a}$$

$$\sigma_{\mathrm{GP}}^2(\mathbf{x}_*) = c_* - \mathbf{k}_*^T(\mathbf{K} + \sigma_e^2\mathbf{I}_n)^{-1}\mathbf{k}_*. \tag{2b}$$

We see that GPs, apart from providing predictions $\mu_{\mathrm{GP}*}$ for a given test input, also have a natural way of assessing the uncertainty of said predictions through the predictive variance (error bars) $\sigma_{\mathrm{GP}*}^2$. The hyperparameters $\boldsymbol{\theta} = [\sigma, \sigma_e]$ to be tuned in the GP determine the width of the SE kernel function and the model noise parameter. There are various ways to learn or infer the hyperparameters, including marginal log-likelihood maximization [22], simple grid search for least squares minimization, or even recent combined strategies [30]. In this paper, we learn $\boldsymbol{\theta}$ using the so-called *pseudo*likelihood [22], the motivation and details of which will be explained in the following.

B. Joint Gaussian Process Regression

Let us now assume that the data set $\mathcal{D}_n$ is formed by two disjoint sets: one set of r real data pairs, $\mathcal{D}_r = \{(\mathbf{x}_i, y_i)\}_{i=1}^r$, and one set of s RTM-simulated pairs $\mathcal{D}_s = \{(\mathbf{x}_i, y_i)\}_{i=r+1}^n$, so that $n = r+s$ and $\mathcal{D}_n = \mathcal{D}_r \cup \mathcal{D}_s$. In the matrix form, we have $\mathbf{X}_r \in \mathbb{R}^{r\times d}$, $\mathbf{y}_r \in \mathbb{R}^{r\times 1}$, $\mathbf{X}_s \in \mathbb{R}^{s\times d}$, and $\mathbf{y}_s \in \mathbb{R}^{s\times 1}$, containing all the inputs and outputs of $\mathcal{D}_r$ and $\mathcal{D}_s$, respectively. Finally, the $n \times 1$ vector $\mathbf{y}$ contains all the n outputs, sorted with the real data first, followed by the simulated data.

A naive approach to incorporating the information of the RTM would be to simply train a regular GP on the data set $\mathcal{D}_n$, not allowing the model to differentiate between data sources. This would accomplish our objective that prediction ought to be guided by simulated data in the regions where real data are scarce.[1] We know, however, that the distributions of the two data sets probably are not identical, and since we aim to predict points belonging to the "real" distribution, we suffer the problem that the auxiliary data might confuse our predictions in regions where we actually possess sufficient real data.

In order to address this problem, a hyperparameter is added to the model, which controls how much the simulated data contribute to prediction. The altered covariance function takes the following form:

$$\boldsymbol{C}_n = \mathbf{K} + \sigma_e^2\mathbf{V}, \quad \mathbf{V} = \mathrm{diag}\big(\underbrace{1, \ldots, 1}_{r}, \underbrace{\gamma^{-1}, \ldots, \gamma^{-1}}_{s}\big) \tag{3}$$

where $\mathbf{K}$ is now an $(r+s) \times (r+s)$ matrix, similar to the formulation of Bonilla *et al.* [31], where we shall call γ the *trust parameter*. It has the straightforward interpretation that it represents the modeled noise-variance in the simulated data, relative to that of the real data, e.g., a model of the form [see (1)]

$$y_i = f(\mathbf{x}_i) + e_i, \quad e_i \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix} \sigma_e^2 & \text{if } i \le r \\ \sigma_e^2/\gamma & \text{if } i > r \end{matrix}\right). \tag{4}$$

We can also consider (2a) written on the kernel smoother form, $\mu_{\mathrm{GP}}(\mathbf{x}_*) = \mathbf{k}_*^T\boldsymbol{\alpha}$. A low trust parameter quenches the components of $\boldsymbol{\alpha}$ pertaining to the simulated data points, thus damping their influence on prediction. We derive a discriminative alternative formulation to this probabilistic perspective of JGP in Appendix A and a multisource formulation to deal with multiple data sets in Appendix B.

C. Learning the Hyperparameters

In this paper, we want to make predictions with respect to the distribution of the real data, so inferring hyperparameters must be done in accordance with this. The common scheme of

[1] This is due to the covariance function defined by the SE kernel, resulting in a high covariance of points that are close in input space.

marginal likelihood maximization [29] is not effective, because it attempts to maximize the likelihood of all data points simultaneously. We, therefore, propose to maximize the leave-one-out (LOO) likelihood, also known as pseudolikelihood [22], allowing us to maximize the likelihood of all data points but the simulated ones. This is reminiscent of the work by Leen *et al.* [32], who construct a focused *model*, where we, in this paper, perform focused *inference*.

The predictive probability of a single training data point conditioned on the remaining data is a normal distribution determined by (2), using all data points but the ith. Thus, the predictive log-likelihood leaving out training point i can be expressed as

$$\log p(y_i|\mathbf{X}_{-i}, \mathbf{y}_{-i}, \boldsymbol{\theta}) = -\frac{1}{2}\log 2\pi\sigma_i^2 - \frac{(y_i - \mu_i)^2}{\sigma_i^2}.$$

From this, we can construct the LOO likelihood by summing over each data point and fit the hyperparameters to maximize it. We modify this approach here, by only summing over the real data points

$$L_{\text{LOO}}(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \sum_{i=1}^{r} \log p(y_i|\mathbf{X}_{-i}, \mathbf{y}_{-i}, \boldsymbol{\theta}). \tag{5}$$

In computing r different predictive means and variances, it appears that we have to invert r slightly different covariance matrices. Luckily, there is a way around this very computationally inefficient approach, which involves simply computing the inverse of the complete covariance matrix [33]. Instead of using (2) a total of r times to evaluate the likelihood function, the following equations may be used:

$$\mu_i = y_i - \frac{[\mathbf{K}^{-1}\mathbf{y}]_i}{[\mathbf{K}^{-1}]_{ii}}, \quad \sigma_i^2 = \frac{1}{[\mathbf{K}^{-1}]_{ii}} \tag{6}$$

where $[\cdot]_i$ denotes the ith element of a vector and $[\cdot]_{ii}$ is the ith diagonal element of a matrix.

D. Joint Gaussian Processes Exemplified

Let us illustrate the solution of the JGP with a toy example. In Fig. 1, we include an illustrative example with real training points (subscript r) covering the range $[-0.6, +0.4]$, and simulated training points (subscript s) in the range $[-1, +1]$. Data were generated from the latent function in black

$$f(x) = b + \exp(-x)\sin(2\pi x) + \epsilon \tag{7}$$

and buried in noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 0.3$. We show the predictive mean of three GP models: one model trained on real data (red curve) and one using real and simulated data together indiscriminitavely (green curve)—these models will be referred to as GP_r and GP_{r+s}, respectively—and finally, the JGP, also using both the data types (magenta curve). We assumed an SE covariance function and learned the optimal hyperparameters with the proposed LOO scheme.

We observe three different regions in the figure. Below $x = -0.6$, we do not have real measurements; hence, the GP_r provides poor estimates, while both the GP_{r+s} and the JGP model provide better fits to the generating function. At the center, $[-0.6, +0.4]$, we have a very accurate view

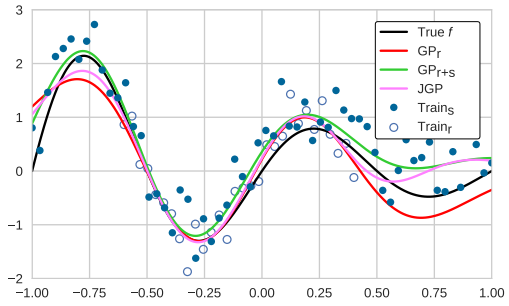


Fig. 1. Example of a JGP in practice.

of the latent function by all methods. For $x > 0.4$, we do not have real training samples neither, so we observe the same behavior as for low values: the GP_r performs poorly revealing a strong bias, and the JGP model fits the observations better than GP_{r+s}; the latter does not weigh the real data points sufficiently high in the overall solution. As commented before, the JGP can distinguish between real and simulated data, and weighs their information differently. This is especially convenient when predicting outside a data-rich, well-represented region, and can be intuitively seen as an "extrapolation" capability of the method.

III. Data Collection

This section is devoted to describing the data used in the experiments. We describe the ground (*in situ*) data set, the remote sensing images acquired over the study areas, and the simulations conducted using PROSAIL.

A. Remote Sensing and Ground Data

The remote sensing and ground data used in this paper were obtained in the framework of the ERMES project [17]. ERMES has developed an agromonitoring system based on the assimilation of Earth observation and *in situ* data for crop modeling solutions for rice monitoring. In this framework, nondestructive ground LAI data were acquired within rice fields in Spain, Italy, and Greece (see Fig. 2) during the 2015 and 2016 European rice seasons. The field campaigns were conducted from the very beginning of rice emergence (early June) up to the maximum rice green LAI development (mid-August), and the temporal frequency of the measurements was approximately ten days. This allowed for a multitemporal database of *in situ* LAI data covering the main phenological rice stages. The sampling was achieved selecting elementary sampling units (ESUs) with different rice varieties and sowing dates in order to cover as much as possible the variability of the study areas, and the locations of the ESUs were far from the field borders. The same sampling scheme was adopted over each ESU, following the guidelines and recommendations of the Validation of Land European Remote sensing Instruments protocol. In addition, the center of the ESU was geolocated to associate the mean LAI measurement with the corresponding satellite spectra.

LAI estimates were acquired in all three countries with smartphones using a dedicated smartphone app called PocketLAI [34], which was previously used in combination with

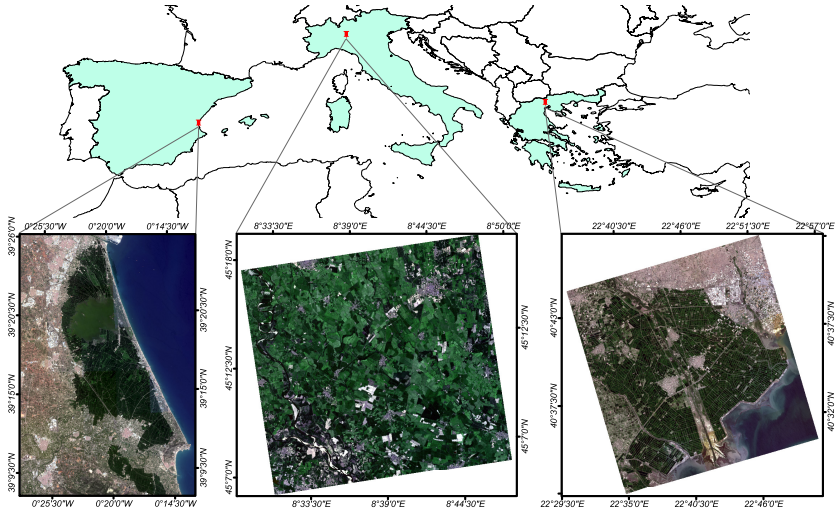


Fig. 2. Study areas: Landsat 8 OLI surface reflectance RGB composites of (Left) Spanish, (Middle) Italian, and (Right) Greek study areas acquired on August 3, 2015, July 18, 2016, and August 25, 2016, respectively.

GPs [28]. PocketLAI uses both the smartphone's accelerometer and camera to acquire images at 57.5° below the canopy and computes LAI through an internal segmentation algorithm [34]. Specifically, over rice fields, we have recently shown that LAI measurements taken with PocketLAI align well with other traditional acquisition instrumentation, such as plant canopy analyzers and digital cameras for hemispherical photography [28], [35]. A range of 18–24 measurements was taken over every ESU in order obtain a statistically significant mean LAI estimate per ESU.

Besides the aforementioned ground data, in this paper, we used Landsat-8 Operational Land Imager (OLI) and Landsat-7 Enhanced Thematic Mapper (ETM+) surface reflectance imagery. The images were downloaded through the United States Geological Survey, Earth Resources Observation and Science and Center Science Processing Architecture during the 2015 and 2016 rice seasons over the three study areas. The provisional Landsat-8 Surface Reflectance (LaSRC) [36] and the Landsat-7 ETM+ Landsat Ecosystem Disturbance Adaptive Processing System products (at 30-m spatial resolution) were used as inputs to retrieve Landsat-7/8 LAI estimates. The Landsat-7 Surface Reflectance/LaSRC spectral channels were filtered to relate only the blue (B), green (G), red (R), near infrared (NIR), and the two short-wave infrared (SWIR1 and SWIR2) bands with the ground LAI measurements in the retrieval process. Images were available every 16 days in Italy and Greece. On the other hand, since the Spanish rice area lies in two Landsat paths within the same row, the temporal resolution of the images is increased up to seven and nine days.

B. RTM Simulations

In this paper, we simulated surface reflectance data of the selected study sites with the PROSAIL RTM. PROSAIL is the most widely used RTM in the last 20 years in remote sensing studies [37]. PROSAIL mimics canopy reflectance using the turbid medium assumption (i.e., assuming the canopy as a turbid medium for which leaves are randomly distributed), which is particularly well suited for homogeneous canopies like rice [38], [39]. PROSAIL simulates leaf reflectance from 400 to 2500 nm with a 1-nm spectral resolution as a function of biochemistry and structure of the canopy, its leaves, the background soil reflectance, and the sun-sensor geometry. Leaf optical properties are given by the mesophyll structural parameter (N), leaf chlorophyll (C_{ab}), dry matter (C_m), and water (C_w) contents. The water content was tied to the dry matter content ($C_w = C_m \times C_{wREL}/(1 - C_{wREL})$) assuming that green leaves have a relative water content (C_{wREL}) varying within a relatively small range [16]. At the canopy level, PROSAIL is characterized by the LAI, the average leaf angle (ALA) inclination and the hotspot parameter (Hotspot). In our experiments, the PROSAIL was run in the forward mode for building a simulated data set (2000 pairs of Landsat-7/8 spectra and associated LAI), which was used for training purposes. In addition, a multiplicative brightness parameter (β_s) was applied to spectral rice background signatures (flooded and dry soil) to represent different background reflectance types [16], [40]. The system geometry was described by the solar zenith angle (θ_s), view zenith angle (θ_v), and the relative azimuth angle between both the angles ($\Delta\Theta$). The distributions for the system geometry were randomly generated based on information in imagery metadata.

It is worth mentioning that in the case of simulating rice crops at high-resolution, subpixel nonvegetated areas located in the borders of rice fields, patches of bare/flooded soil, small water stripes, and channels must be represented in the PROSAIL simulation [39]. Hence, in order to account for these mixed conditions, we represented the pixels as a linear mixture of vegetation ($vCover$) and bare/flooded soil ($1 - vCover$) spectra. A linear spectral mixing model was assumed for the sake of simplicity.

The leaf and canopy variables, as well as the soil brightness and the $vCover$ parameter were randomly generated following

TABLE I
DISTRIBUTION OF THE CANOPY, LEAF, AND SOIL PARAMETERS USED IN THIS PAPER FOR SIMULATION WITH THE PROSAIL RTM

Parameter		Min	Max	Mode	Std	Type
Canopy	LAI (m^2/m^2)	0	10	3.5	4.5	Gaussian
	ALA (°)	30	80	60	20	Gaussian
	Hotspot	0.1	0.5	0.2	0.2	Gaussian
	vCover	0.5	1	1	0.2	Trunc. Gaussian
Leaf	N	1.2	2.2	1.5	0.3	Gaussian
	C_{ab} ($\mu g \cdot cm^2$)	20	90	45	30	Gaussian
	C_{dm} ($g \cdot cm^2$)	0.003	0.011	0.005	0.005	Gaussian
	C_{wREL}	0.6	0.8	-	-	Uniform
Soil	β_s	0.3	1.2	0.9	0.25	Gaussian

the parametrization in [39] and [41] in order to constrain the behavior of the model to Mediterranean rice areas (see Table I). In particular, a spectral library of underlying rice background (flooded and dry) signatures was used to obtain multitemporal LAI retrievals robust against changes in the background condition related to water management.

C. On the Data Distributions

Blending *in situ* and simulated data requires a careful evaluation of the representativity of the data. When the distribution of the RTM-simulated data does not match the characteristics observed in real data, models using simulated data can be prone to error, because learning good hyperparameters becomes a difficult task. Intuitively, the JGP model tries to learn the relative relevance of both the sources of information, which is impossible when data sets do not follow the same (or a similar) distribution. In this case, the model is most likely to disregard the information of the simulated data completely. It is important to remember that generating simulated data, through choosing sensible parameter ranges in PROSAIL that is difficult, requires expert knowledge, and is scenario-dependent. Scatterplots in Fig. 3 show the distributions represented in the space of NDVI-vs-LAI for all sites and acquisition campaigns. These joint distributions suggest that the simulated points (in blue) cover regions of the greenness-LAI space efficiently for Spain and Italy, but cannot match the wide noise levels and variance observed in the real Greece data distributions, regardless the campaign. As described in Section IV, this has implications in the obtained results.

IV. EXPERIMENTAL RESULTS

This section presents the experimental results obtained with the JGP model. We first evaluate the important issues of bias and noise variance in synthetic data distributions, and how the JGP deals with them. Then, the empirical evidence of performance in two real experiments is given. First, we evaluate LAI prediction from Landsat images in all three sites and the two campaigns, and finally, we analyze the performance in an extrapolation scenario.

A. Robustness to Bias and Noise

We use the same generating function as in Section II-D to illustrate the capabilities of the JGP to deal with systematic bias, and varying noise regimes [respectively, b and σ in (7)]. In particular, we generated "real" data sets on the restricted interval shown in Fig. 1, and "simulated" data sets on the wider interval of $[-1; 1]$ with different levels of white noise variance σ^2_{sim} and values of an added bias b_{sim}. The test data were generated in the same way as the real training data, but over the entire interval $[-1; 1]$, imitating a case of extrapolation where real data are unavailable in some domains, but simulated data of varying quality can be obtained across the whole representation space. We compare the performance of the JGP with the naive approach of training a regular GP on the combined data sets (GP_{r+s}), as well as that of a GP trained only on simulated (GP_s) or real data only (GP_r). The JGP hyperparameters are found by optimizing pseudolikelihood over the real data, as described in Section II-C. The other methods also maximize pseudolikelihood, however, over all their data, for optimal comparison. These models might as well use standard marginal likelihood [29] maximization.

From Fig. 4, we can see the obvious result that if the simulated data are perfect, i.e., just points from the underlying damped sine, all methods that use the simulated data are performing better than the GP trained only on real data. Conversely, if the simulated data are very dissimilar to the real, it is better not to use it at all. Depending on how the distribution of the two data sets diverges, there is a risk that the simulated data confuse the regression. We see that the JGP is the approach to incorporating simulated data, which, roughly speaking, best handles a deterioration in the quality of these data points, be it through noisy regimes or systematic bias. The main takeaway here is that, as it is uncertain to know in advance how helpful and realistic simulations are, the JGP presents a safe way to incorporate physical knowledge about the inverse problem at hand.

B. LAI Retrieval From Landsat Images

In this section, we assess the performance of the JGP and compare it with other ways of including simulated data when attempting to solve the inversion problem: the GP_{r+s} and the GP_s approach. To this end, we use each of the six data sets collected through the campaigns in the respective countries (Spain, Greece, and Italy) and years (2015 and 2016). Root mean squared error (RMSE) is computed using a tenfold cross-validation scheme. During each fold, the amount of simulated data used by the JGP and GP_{r+s} is gradually increased in order to study how the ratio of simulated-to-real data points, call it p, affects method's performance. The full 2000 simulated points are used for the GP_s, since they are generated by PROSAIL to represent the target area as well as possible. Finally, the experiment is repeated 50 times to get stable results.

The averaged RMSE as a function of the included simulated data is shown in Fig. 5. We observed rather different behaviors for the different data sets and scenarios. There are cases where γ is fitted to a value close to 0, i.e., the JGP ignores the added data and simply follows the GP_s baseline. For the data sets where this is not the case, we observe that relatively a little simulated data are needed ($p \sim 0.5$) to produce an effect. This is worth noting as the inversion of the kernel matrix,

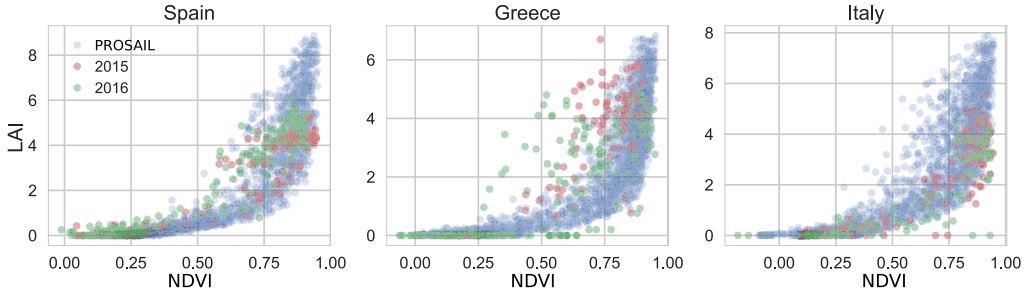


Fig. 3. Scatterplots in the NDVI-LAI representation space of the real and RTM-simulated data for all sites and acquisition campaigns (2015 and 2016).

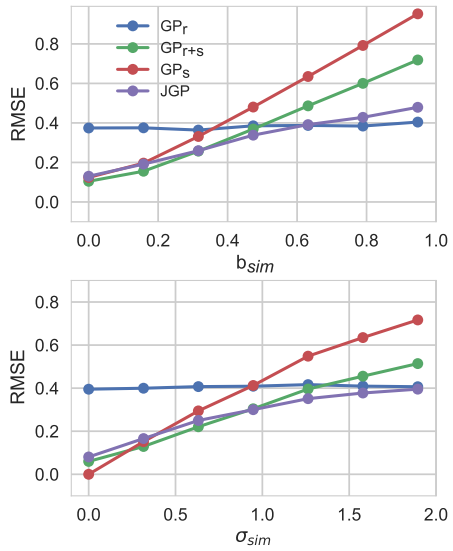


Fig. 4. Performance of different schemes for including simulated data in a toy example where the quality of the secondary data source is varied.

needed to train the JGP, scales in time complexity with the number of samples cubed, $\mathcal{O}(n^3) = \mathcal{O}((r+s)^3)$. This is equally true of the different methods used, regardless of the likelihood used. Furthermore, for predicting m new points, we have $\mathcal{O}((r+s)m)$.

In the case of Greece 2015, an average increase in RMSE is observed, whose percentagewise is around $\sim$1%. In Spain 2015 and Greece 2016, a decrease in the RMSE of around $\sim$5% can be observed. Interestingly, we see that the naive inclusion of simulated data (the GP_{r+s} scheme) results in a general increase in error, except for the case of Greece 2016. This might be explained through the results shown in Fig. 4, where GP_{r+s} performs slightly better than the JGP approach when the simulated data are of high quality.

The approach of using only simulated data for predicting LAI, although it has been shown to aptly capture the temporal evolution of vegetation [39], shows considerable predictive error, visually distorting the results of Fig. 5. The performance of the GP_s method is, therefore, instead given in Table II. Comparing with the baseline of Fig. 5, we see that it suffers

TABLE II
PERFORMANCE OF GP_s METHOD

RMSE GP_s	Spain	Greece	Italy
2015	0.92	1.87	1.07
2016	1.09	1.32	1.20

from a constant increase in RMSE of at least 25%. This underlines the point that, although RTM-simulated data reflects the physical relation between the input and the output (spectrum and LAI), it struggles to mimic *in situ* data.

C. LAI Retrieval in Extrapolation Scenario

In this last experiment, we demonstrate the "far extrapolation use-case" for the JGP method. The experiment in Section IV-B, in practice, creates small "holes" in the real data distribution by removing a tenth for testing according to the tenfold cross-validation scheme. These holes in the representation space might then be filled with simulated data. The natural use, however, is the one where extrapolation is necessary for a region where no training data exist, but where a physical model might generate physically meaningful data points. Such scenarios might come about due to cloud coverage, unsystematic sampling in through growing seasons, erroneous *in situ* measurements, and so on.

A way to imitate such a scenario is to locate the green band median $\tilde{x}_G$ of the *in situ* data and split it 50–50 such that the training and test data, respectively, have low and high values in the green band. This corresponds to the rather unrealistic case, where sampling takes place only in the beginning of the year. We also used the upper quartile of the green band to perform a 75–25 split of training-test data. A p of 1 was chosen for the JGP and GP_{r+s}, while the GP_s was trained on all 2000 RTM-simulated data points as before. The gain in performance in such a scenario is shown in Table III, showing generally large RMSE reductions for all methods compared with GP_r, although the baseline is unreasonably high. In this experiment, the JGP does the best when it during the training phase fits a high trust parameter, i.e., it deems that the RTM-data are predictive of the real data. In Spain, this does not appear to be occurring. It is, however, the only method that does not perform worse than GP_r on any data set.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

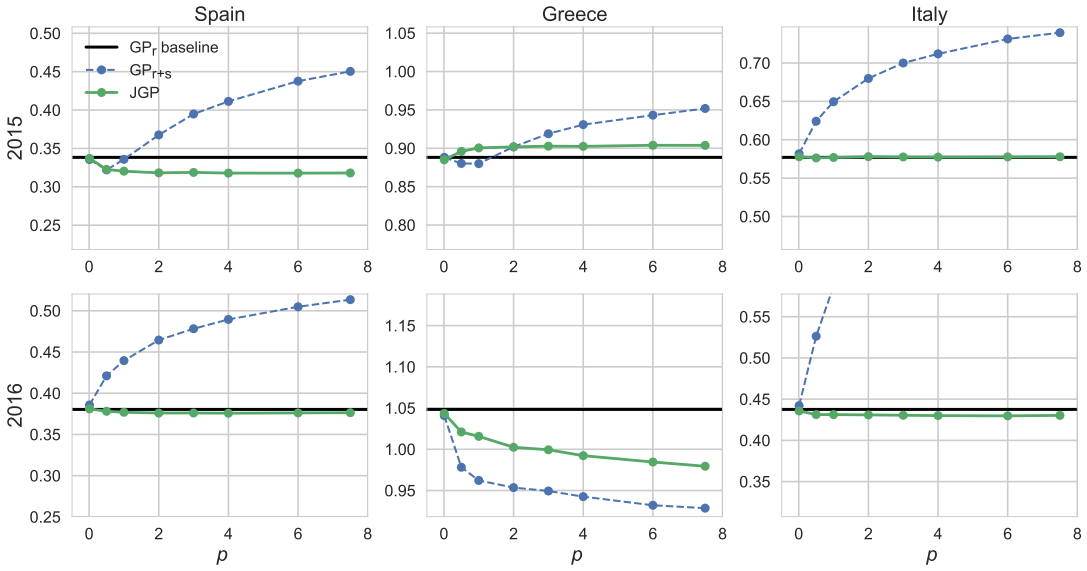


Fig. 5. Performance comparison (RMSE) for different ways of including simulated data. The JGP and the regular GP, trained on a data set of real and simulated data pooled together, are compared with the base line of the GP trained exclusively on real data. RMSE is shown for the different sites, campaign dates, and simulated-to-real data ratios. As the scale is constant over the plots for better comparison, it was omitted from the plot in Italy 2016 how the GP_{r+s} RMSE monotonically increases and reaches 0.85 for $p = 8$.

TABLE III

PERFORMANCE OF THE GP_r, GP_{r+s}, GP_s, AND JGP METHODS WHEN DIVIDING THE REAL DATA SO THAT TEST AND TRAINING DATA ARE WELL-SEPARATED DOMAINS. (TOP AND BOTTOM ROWS) RESULTS FROM THE 50–50 AND 75–25 PARTITION SCHEMES, RESPECTIVELY (SEPARATED BY THICK HORIZONTAL LINE)

RMSE: GP_r/GP_{r+s}/GP_s/JGP		Spain	Greece	Italy
50-50	2015	3.78 / 1.26 / 1.28 / 3.12	3.05 / 1.30 / 2.41 / 2.28	1.82 / 1.63 / 1.50 / 1.56
	2016	4.31 / 1.65 / 1.50 / 2.92	2.90 / 1.78 / 1.87 / 2.69	1.91 / 1.36 / 1.70 / 0.77
75-25	2015	2.30 / 0.914 / 1.29 / 1.72	1.80 / 1.29 / 2.85 / 1.31	1.16 / 1.18 / 1.77 / 0.961
	2016	2.44 / 0.94 / 1.59 / 2.43	3.33 / 1.89 / 1.73 / 2.83	1.193 / 1.64 / 2.22 / 0.77

V. CONCLUSION

This paper introduced a method based on GPs for biophysical parameter retrieval. To the best of our knowledge, this is the first statistical nonparametric model blending *in situ* measurements and RTM-simulations. The model allows for the combination of *in situ* data and simulated data generated by an RTM code. The formulation of JGP only incorporates one additional tradeoff hyperparameter that learns the relative importance of real and simulated data, and is related to the specific noise variance in each data set. In the training phase, pseudolikelihood is maximized with respect to the real data only, which was shown to be a safe way of including simulated data. We studied the model in terms of accuracy, robustness to bias, and noise regimes, and performed simulations in high missing data regimes.

We illustrated the performance in the particular case of estimating LAI using Landsat images and simulated data from PROSAIL. Noticeable gains in accuracy were obtained in general. The model exploits the space coverage of RTMs in regions where real data scarcity hampers performance, while at the same time respecting the information provided by real data. Given the wide applicability of the JGP model, we foresee applications of the model in domains other than vegetation monitoring where a few real data can be acquired yet a mechanistic model is available. It is also worth noting that incorporation of RTM-simulated data is not restricted to GPs, i.e., other regression methods could benefit from this as well.

Future work is tied to study the capabilities of the model for transportability across space and time simultaneously. For that, we plan to incorporate anisotropic and invariant kernels. In this sense, manifold alignment could benefit the model, for example, by projecting simulated data distributions into the real one before doing the regression. This in principle should reduce the problems of mismatching and representativity of the simulations. Finally, it is worth noting that the JGP model is easily extended to deal with multiset scenarios, as shown in Appendix B. Therefore, different campaigns, sites, and teams could receive different trust hyperparameters in the model. This actually relates to the field of multitask learning, which has received a little attention in remote sensing data processing and for classification problems only.

APPENDIX A
LEAST SQUARES JGP FORMULATION

Let us derive a discriminative alternative formulation to the probabilistic perspective of JGP presented in Section II. We will follow the same rationale as in standard least squares regression with kernel methods [27]. We are given input data matrices $\mathbf{X}_r \in \mathbb{R}^{r\times d}$, $\mathbf{X}_s \in \mathbb{R}^{s\times d}$, and the corresponding target vectors $\mathbf{y}_r$ and $\mathbf{y}_s$. We can define the collectively grouped

data as $\mathbf{X}_n \in \mathbb{R}^{n\times d}$ and $\mathbf{y}_n$. Let us now define two kernel feature mappings $\boldsymbol{\phi}_r$ and $\boldsymbol{\phi}_s$ that map real and simulated data, respectively, to a Hilbert feature space, $\mathcal{H}$, which may be in principle of higher (possibly infinite) dimensionality than d, i.e., $H = dim(\mathcal{H}) \gg d$. We indicate the mapped data matrices as $\boldsymbol{\Phi}_r \in \mathbb{R}^{r\times H}$ and $\boldsymbol{\Phi}_s \in \mathbb{R}^{s\times H}$.

Now let us define the following cost function $\mathcal{L}$ that trades off the prediction errors using real or simulated data, and the standard regularization parameter:

$$\mathcal{L} = \|\mathbf{y}_r - \boldsymbol{\Phi}_r\mathbf{w}\|^2 + \lambda_1\|\mathbf{y}_s - \boldsymbol{\Phi}_s\mathbf{w}\|^2 + \lambda_2\|\mathbf{w}\|^2.$$

Differentiating with respect to $\mathbf{w}$ and equating to zero, we obtain

$$\left(\boldsymbol{\Phi}_r^\top\boldsymbol{\Phi}_r + \lambda_1\boldsymbol{\Phi}_s^\top\boldsymbol{\Phi}_s + \lambda_2\mathbf{I}\right)\mathbf{w} = \boldsymbol{\Phi}_r^\top\mathbf{y}_r + \lambda_1\boldsymbol{\Phi}_s^\top\mathbf{y}_s.$$

Now, by applying the following representer's theorem [42], $\mathbf{w} = [\boldsymbol{\Phi}_r^\top \;\; \boldsymbol{\Phi}_s^\top]\boldsymbol{\alpha} = \boldsymbol{\Phi}_n\boldsymbol{\alpha}$, and premultiplying from the left by $\boldsymbol{\Phi}_n$ yields the solution:

$$\boldsymbol{\Phi}_n\left(\boldsymbol{\Phi}_r^\top\boldsymbol{\Phi}_r + \lambda_1\boldsymbol{\Phi}_s^\top\boldsymbol{\Phi}_s + \lambda_2\mathbf{I}\right)\boldsymbol{\Phi}_n^\top\boldsymbol{\alpha} = \boldsymbol{\Phi}_n\left(\boldsymbol{\Phi}_r^\top\mathbf{y}_r + \lambda_1\boldsymbol{\Phi}_s^\top\mathbf{y}_s\right)$$

which can be expressed solely in terms of kernel matrices as

$$(\mathbf{K}_{nr}\mathbf{K}_{rn} + \lambda_1\mathbf{K}_{ns}\mathbf{K}_{sn} + \lambda_2\mathbf{K}_{nn})\boldsymbol{\alpha} = [\mathbf{K}_{nr}\mathbf{y}_r \quad \lambda_1\mathbf{K}_{ns}\mathbf{y}_s]$$

and then the solution comes in a closed form as

$$\boldsymbol{\alpha} = (\mathbf{K}_{nr}\mathbf{K}_{rn} + \lambda_2\mathbf{K}_{ns}\mathbf{K}_{sn} + \lambda_1\mathbf{K}_{nn})^{-1}\;[\mathbf{K}_{nr}\mathbf{y}_r \quad \lambda_1\mathbf{K}_{ns}\mathbf{y}_s]$$

where the subscripts of the kernel matrices, which come about from the interpretation that the kernel function defines the inner product on the space $\mathcal{H}$, indicate their sizes and the samples involved in their calculation. Note that when $\lambda_1 = 0$, the standard kernel ridge regression (or equivalently the predictive mean for the standard GP) is obtained; otherwise, λ_1 acts as an extra regularization term accounting for the relative importance of the real and the simulated data points.

Note that by convenient term grouping, by defining the diagonal matrix $\mathbf{V} = \text{diag}(1, \ldots, 1, \lambda_1, \ldots, \lambda_1)$, and by identifying the regularization term as the noise term in the probabilistic view of GPs (i.e., $\lambda_2 = \sigma_e^2$), we reach the equivalent JGP model in Section II-B, which yields the simpler solution of the predictive mean with $\boldsymbol{\alpha} = (\mathbf{K}_{nn} + \sigma_e^2\mathbf{V})^{-1}\mathbf{y}$. Unfortunately, with a pure discriminative approach, one misses the probabilistic interpretation of the model and hyperparameters, and restricts oneself to mean predictions only.

APPENDIX B
MULTISOURCE JGP FORMULATION

The JGP formulation as presented in this paper assumes access to two data sets only: one coming from a "main" distribution according to which we wish to make predictions (real *in situ* data in our experiments), and one coming from an "auxiliary" distribution (simulated data from an physical RTM model in our case). We could also generalize the formulation and assume that we access m such auxiliary data sets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_m\}$, each holding a different number of data points $\{s_1, s_2, \ldots, s_m\}$. The JGP can easily be extended to this multisource scenario by fitting a trust parameter to each data set. The $\mathbf{V}$ matrix of the covariance function in (3) simply becomes

$$\mathbf{V} = \text{diag}\big(\underbrace{1, \ldots, 1}_{r}, \underbrace{\gamma_1^{-1}, \ldots, \gamma_1^{-1}}_{s_1}, \ldots\ldots, \underbrace{\gamma_m^{-1}, \ldots, \gamma_m^{-1}}_{s_m}\big) \quad (8)$$

and the same solution applies. Note the relation of this multisource JGP to multitask formulations previously presented in remote sensing data classification [43].

ACKNOWLEDGMENT

The authors would like to thank the Institute for Electromagnetic Sensing of the Environment, the Cereal Institute of DEMETER, and the Aristotle University of Thessaloniki for providing the Italian and Greek field data acquired under the ERMES project.

REFERENCES

[1] R. Snieder and J. Trampert, *Inverse Problems in Geophysics*. Vienna, Austria: Springer, 1999, pp. 119–190.

[2] T. Hilker, N. C. Coops, M. A. Wulder, T. A. Black, and R. D. Guy, "The use of remote sensing in light use efficiency based models of gross primary production: A review of current status and future requirements," *Sci. Total Environ.*, vol. 404, nos. 2–3, pp. 411–423, 2008.

[3] R. H. Whittaker and P. L. Marks, "Methods of assessing terrestrial productivty," in *Primary Productivity of the Biosphere*. Berlin, Germany: Springer, 1975, pp. 55–118.

[4] H. K. Lichtenthaler, "Chlorophylls and carotenoids: Pigments of photosynthetic biomembranes," *Methods Enzymol.*, vol. 148, pp. 350–382, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0076687987480361

[5] S. Jacquemoud, C. Bacour, H. Poilvé, and J.-P. Frangi, "Comparison of four radiative transfer models to simulate plant canopies reflectance: Direct and inverse mode," *Remote Sens. Environ.*, vol. 74, no. 3, pp. 471–481, 2000.

[6] W. Verhoef and H. Bach, "Simulation of hyperspectral and directional radiance images using coupled biophysical and atmospheric radiative transfer models," *Remote Sens. Environ.*, vol. 87, no. 1, pp. 23–41, 2003.

[7] S. Liang, *Advances in Land Remote Sensing: System, Modeling, Inversion and Applications*. Berlin, Germany: Springer, 2008.

[8] G. Camps-Valls, D. Tuia, L. Gómez-Chova, and J. Malo, Eds., *Remote Sensing Image Processing*. San Rafael, CA, USA: Morgan & Claypool, Sep. 2011.

[9] J. Verrelst *et al.*, "Optical remote sensing and the retrieval of terrestrial vegetation bio-geophysical properties—A review," *ISPRS J. Photogram. Remote Sens.*, vol. 108, pp. 273–290, Oct. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271615001422

[10] T. J. Ulrych, M. D. Sacchi, and A. Woodbury, "A Bayes tour of inversion: A tutorial," *Geophysics*, vol. 66, no. 1, pp. 55–69, 2001.

[11] J. Verrelst, L. Alonso, G. Camps-Valls, J. Delegido, and J. Moreno, "Retrieval of vegetation biophysical parameters using Gaussian process techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 5, pp. 1832–1843, May 2012.

[12] H. Fang and S. Liang, "A hybrid inversion method for mapping leaf area index from MODIS data: Experiments and application to broadleaf and needleleaf canopies," *Remote Sens. Environ.*, vol. 94, no. 3, pp. 405–424, 2005.

[13] H. Fang and S. Liang, "Retrieving leaf area index with a neural network method: Simulation and validation," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 2052–2062, Sep. 2003.

[14] G. Camps-Valls, J. Verrelst, J. Munoz-Mari, V. Laparra, F. Mateo-Jimenez, and J. Gomez-Dans, "A survey on Gaussian processes for earth-observation data analysis: A comprehensive investigation," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 58–78, Jun. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7487896/

[15] J. Verrelst *et al.*, "Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for Sentinel-2 and -3," *Remote Sens. Environ.*, vol. 118, pp. 127–139, Mar. 2012.

[16] F. Baret *et al.*, "LAI, fAPAR and fCover CYCLOPES global products derived from VEGETATION: Part 1: Principles of the algorithm," *Remote Sens. Environ.*, vol. 110, no. 3, pp. 275–286, 2007.

[17] L. Busetto *et al.*, "Downstream services for Rice crop monitoring in europe: From regional to local scale," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/document/7898821/?reload=true

[18] K. F. Wallis, "Combining forecasts—Forty years later," *Appl. Financial Econ.*, vol. 21, nos. 1–2, pp. 33–41, 2011.

[19] R. F. Bordley, "The combination of forecasts: A Bayesian approach," *J. Oper. Res. Soc.*, vol. 33, no. 2, pp. 171–174, 1982.

[20] S. L. Scott, A. W. Blocker, F. V. Bonassi, H. A. Chipman, E. I. George, and R. E. McCulloch, "Bayes and big data: The consensus Monte Carlo algorithm," *Int. J. Manage. Sci. Eng. Manage.*, vol. 11, no. 2, pp. 78–88, 2016.

[21] D. Luengo, L. Martino, V. Elvira, and M. Bugallo, "Efficient linear combination of partial Monte Carlo estimators," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4100–4104.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York, NY, USA: MIT Press, 2006.

[23] J. Verrelst, L. Alonso, G. Camps-Valls, J. Delegido, and J. Moreno, "Retrieval of canopy parameters using Gaussian processes techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 5, pp. 1832–1843, 2012.

[24] J. Verrelst, L. Alonso, J. P. Rivera, J. Moreno, and G. Camps-Valls, "Gaussian process retrieval of chlorophyll content from imaging spectroscopy data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 867–874, Apr. 2013.

[25] M. Lázaro-Gredilla, M. K. Titsias, J. Verrelst, and G. Camps-Valls, "Retrieval of biophysical parameters with heteroscedastic Gaussian processes," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 4, pp. 838–842, Apr. 2014.

[26] G. Tramontana *et al.*, "Predicting carbon dioxide and energy fluxes across global FLUXNET sites with regression algorithms," *Biogeosci. Discussions*, vol. 13, no. 14, pp. 4291–4313, 2016. [Online]. Available: http://www.biogeosciences-discuss.net/bg-2015-661/

[27] G. Camps-Valls and L. Bruzzone, Eds., *Kernel Methods for Remote Sensing Data Analysis*. Chichester, U.K.: Wiley, Dec. 2009.

[28] M. Campos-Taberner *et al.*, "Mapping leaf area index with a smartphone and Gaussian processes," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2501–2505, Dec. 2015.

[29] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[30] L. Martino, V. Laparra, and G. Camps-Valls, "Probabilistic cross-validation estimators for Gaussian process regression," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 1–5.

[31] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 153–160.

[32] G. Leen, J. Peltonen, and S. Kaski, "Focused multi-task learning in a Gaussian process framework," *Mach. Learn.*, vol. 89, nos. 1–2, pp. 157–182, 2012.

[33] S. Sundararajan and S. Keerthi, "Predictive approaches for choosing hyperparameters in Gaussian processes," *Neural Comput.*, vol. 13, no. 5, pp. 1103–1118, May 2001.

[34] R. Confalonieri *et al.*, "Development of an app for estimating leaf area index using a smartphone. Trueness and precision determination and comparison with other indirect methods," *Comput. Electron. Agricult.*, vol. 96, pp. 67–74, Aug. 2013.

[35] M. Campos-Taberner *et al.*, "Multitemporal monitoring of plant area index in the valencia Rice district with PocketLAI," *Remote Sens.*, vol. 8, no. 3, p. 202, 2016.

[36] E. Vermote, C. Justice, M. Claverie, and B. Franch, "Preliminary analysis of the performance of the Landsat 8/OLI land surface reflectance product," *Remote Sens. Environ.*, vol. 185, pp. 46–56, Nov. 2016.

[37] S. Jacquemoud *et al.*, "PROSPECT + SAIL models: A review of use for vegetation characterization," *Remote Sens. Environ.*, vol. 113, pp. S56–S66, Sep. 2009.

[38] R. Darvishzadeh, A. A. Matkan, and A. D. Ahangar, "Inversion of a radiative transfer model for estimation of Rice canopy chlorophyll content using a lookup-table approach," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 4, pp. 1222–1230, Aug. 2012.

[39] M. Campos-Taberner *et al.*, "Multitemporal and multiresolution leaf area index retrieval for operational local Rice crop monitoring," *Remote Sens. Environ.*, vol. 187, pp. 102–118, Dec. 2016.

[40] M. Claverie, E. F. Vermote, M. Weiss, F. Baret, O. Hagolle, and V. Demarez, "Validation of coarse spatial resolution LAI and FAPAR time series over cropland in southwest France," *Remote Sens. Environ.*, vol. 139, pp. 216–230, Dec. 2013.

[41] M. Campos-Taberner *et al.*, "Exploitation of SAR and optical sentinel data to detect Rice crop and estimate seasonal dynamics of leaf area index," *Remote Sens.*, vol. 9, no. 3, p. 248, 2017.

[42] F. Riesz and B. S. Nagy, *Functional Analysis*. New York, NY, USA: Frederick Unger, 1955.

[43] J. M. Leiva-Murillo, L. Gomez-Chova, and G. Camps-Vall, "Multitask remote sensing data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 151–161, Jan. 2012.

Daniel Heestermans Svendsen received the B.Sc. degree in physics and nanotechnology and the M.Sc. degree in mathematical modeling and computation from the Technical University of Denmark, Lyngby, Denmark, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Image and Signal Processing Group, Universitat de València, València, Spain.

His research interests include GPs and incorporation of physical knowledge in statistical methods.

Luca Martino was born in Palermo, Italy, in 1980. He received the M.Sc. degree in electronic engineering from the Politecnico di Milano, Milan, Italy, and the Ph.D. degree in statistical signal processing from the Universidad Carlos III de Madrid, Madrid, Spain, in 2011.

He spent two years in the Department of Statistics, University of Helsinki, Helsinki, Finland. He carried out a post-doctoral research at the Sao Paulo Research Foundation (FAPESP), São Paulo, Brazil, and at the Universitat de València, València, Spain.

Manuel Campos-Taberner received the bachelor's degree in physics, and the master's and Ph.D. degrees in remote sensing from the Universitat de València, València, Spain, in 2012, 2013, and 2017, respectively.

He took part in the winning team of the 2015 IEEE Geoscience and Remote Sensing Society Data Fusion Contest, and he is currently involved in biophysical parameters retrieval at local and global scales. His research interests include the development and validation of earth observation chains for vegetation monitoring, including field campaigns, image processing, and machine learning regression algorithms for biophysical parameter estimation.

Francisco Javier García-Haro is currently an Associate Professor of earth physics with the Universitat de València, València, Spain. He has been responsible for many research projects related to canopy radiative transfer modeling and retrieval of vegetation properties using satellite, including remote sensing applications such as agrometeorology, land and soil resources, agriculture, and forestry. The scientific production includes 50 papers, over 150 conference proceedings, and numerous technical reports. He is involved in several validation networks and exploitation programs of satellite missions.

Dr. García-Haro was a recipient of several research awards.

Gustau Camps-Valls (M'04–SM'07) received the Ph.D. degree in physics from the Universitat de València, València, Spain, in 2002.

He is currently a Full professor in electrical engineering, and a Coordinator with the Image and Signal Processing Group, Universitat de València. He is interested in the development of machine learning algorithms for geoscience and remote sensing data analysis.

Pattern Recognition 100 (2020) 107103

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Active emulation of computer codes with Gaussian processes – Application to remote sensing

Daniel Heestermans Svendsen[a,*], Luca Martino[b], Gustau Camps-Valls[a]

[a] Image Processing Lab (IPL), Universitat de València, C/ Cat. José Beltrán, 2. Paterna 46980, Spain
[b] Dep. Signal Processing, Universidad Rey Juan Carlos (URJC), Camino del Molino 5, Fuenlabrada 28943, Spain

ARTICLE INFO

Article history:
Received 18 February 2019
Revised 25 October 2019
Accepted 3 November 2019
Available online 13 November 2019

Keywords:
Active learning
Gaussian process
Emulation
Design of experiments
Computer code
Remote sensing
Radiative transfer model

ABSTRACT

Many fields of science and engineering rely on running simulations with complex and computationally expensive models to understand the involved processes in the system of interest. Nevertheless, the high cost involved hamper reliable and exhaustive simulations. Very often such codes incorporate heuristics that ironically make them less tractable and transparent. This paper introduces an active learning methodology for adaptively constructing surrogate models, i.e. *emulators*, of such costly computer codes in a multi-output setting. The proposed technique is sequential and adaptive, and is based on the optimization of a suitable acquisition function. It aims to achieve accurate approximations, model tractability, as well as compact and expressive simulated datasets. In order to achieve this, the proposed Active Multi-Output Gaussian Process Emulator (AMOGAPE) combines the predictive capacity of Gaussian Processes (GPs) with the design of an acquisition function that favors sampling in low density and fluctuating regions of the approximation functions. Comparing different acquisition functions, we illustrate the promising performance of the method for the construction of emulators with toy examples, as well as for a widely used remote sensing transfer code.

1. Introduction

In many areas of science and engineering, systems are analyzed by running computer code simulations which act as convenient approximations of reality. They allow us to simulate many different systems of interest and characterize the involved processes, such as turbulence or energy transfer, and their interactions and relevance. Depending on the body of literature, they are known as physics-based or mechanistic models, or simply *simulators* [1,2]. Two important limitation are associated with simulators. The first, and perhaps the most important problem of these computer codes, is their often high computational cost, which hampers reliable and exhaustive simulations. This limits the representativity of the simulations, which in turn makes numerical or statistical inversion a hard problem. Secondly, since computer codes rely on decades of intensive development and parametrizations, they often include heuristics that improve accuracy but ironically make them less mathematically tractable and transparent.

1.1. Emulation for forward models

In the last decade, a field collectively known as *surrogate modeling* or *emulation* has emerged as an efficient alternative: emulators try to mimic costly computer codes with machine learning models. The field of emulation has received attention from subfields of statistical signal processing and machine learning [3–7]. In order to construct an emulator, we need a simulated dataset which is made by evaluating the computer code in different input points. The problem of choosing these points, for which this paper presents an active learning algorithm [8,9], is treated in different parts of the statistics and machine learning literature. A non-exhaustive overview is given below.

1.2. Related work

The problem at hand is closely related to that of Design of Experiments (DOE), where one seeks a set of input values which best allows one to determine the relationship between inputs and outputs. Between the algorithms that will be reviewed in this section, there are key some differences between types of algorithms that it would be beneficial to clarify first:

- *Sequential vs. non-sequential* refers to whether the algorithm needs to know a priori how many input points to choose. Non-

* Corresponding author.
E-mail addresses: daniel.svendsen@uv.es (D.H. Svendsen), lmartino@ing.uc3m.es (L. Martino), gustau.camps@uv.es (G. Camps-Valls).

https://doi.org/10.1016/j.patcog.2019.107103
0031-3203/

sequential or one-shot algorithms need this information, while sequential algorithms can simply run until some time limit or accuracy criteria is met, which is a favourable property. Between the two approaches lie the batch-sequential algorithms.

- *Continuous vs. discrete sampling* refers to whether an algorithm aims to either choose input points in a continuous space or choose among a finite set of points. A large part of the literature deals with the latter problem, and relies on greedy and MCMC algorithms to choose points according to some criterion. Many of the methods proposed in the literature can be easily adapted from continuous to discrete sampling and vice versa.

Among the most popular criteria are maximum entropy [10], maximizing distance to nearest neighbour [11], and minimizing integrated root mean squared error [12]. These criteria have been implemented for construction of GP emulators in both a sequential and batch-sequential way [13]. An interesting approach in this field is the Bayesian Experimental Design (BED) which assumes a probabilistic model of the observed data and defines a so-called utility function based on the posterior of the model parameters. The approach then aims to maximize the mean of the utility. Recent relevant work can be found in [14,15]. A great deal of DOE methods, even the sequential ones, do not assume the ability to *query* a system, due to the way experiments are carried out.

The field of Active Learning (AL), on the other hand, builds on the premise that we can query a system and thus learn something about it in each iteration [16,17]. Building an emulator sequentially is a problem that fits directly into this category. The algorithms in the AL literature concerned with GP regression often employ criteria based on predictive variance [18] and entropy [19]. Other algorithms are based on triangulation of the input space [20] or gridding [21], followed by a ranking of each triangle or cell. Greedily searching for candidate points which have maximum distance to their nearest neighbours [22] has also proven effective. Furthermore, when the input set is comprised of finite discrete values, interesting criteria like mutual information have been employed with success [23].

1.3. Our contribution: active emulation as a step forward

In this paper, we introduce a methodology for developing efficient machine learning emulators of costly physical models based on active emulation. An active learning framework is developed that *sequentially* chooses informative input points, learning about the underlying function as the algorithm progresses. This active emulation methodology is based on the notion of an *acquisition function* which can be optimized through gradient-based techniques, mirroring approaches in Bayesian Optimization [24]. The goal is to construct an accurate emulator with as few runs of the computer code as possible.

Given a set of initial datapoints, the emulator is built through the online addition of new nodes[1], maximizing the acquisition function at each iteration. The acquisition function is constructed to incorporate (a) geometric information of the costly, analytically intractable function $\mathbf{f}$, and (b) information about the distribution of the current nodes. By using Gaussian processes we can derive both terms analytically, and for multiple outputs at once. The reasoning is that areas of high variability in $\mathbf{f}(\mathbf{x})$ requires the addition of more information, as has also been noted in [21]. In [25] the predictive variance of the gradient norm of a GP is used as a sampling criteria, which is a less straightforward approach than just using the gradient directly as done here. Similarly, regions with a small concentration of nodes requires the introduction of new nodes in order to fill the space (simple exploration, space filling without taking into account the geometrical features of $\mathbf{f}(\mathbf{x})$). We show how to define such an acquisition function in a multi-output setting. Fig. 1 shows an illustrative example of the building blocks of the active emulation methodology presented here.

The developed methodology of constructing emulators is sequential and searches a continuous input-space, leading to emulators that are *accurate*, so they can be taken as a faithful representation of the physical models and codes, *compact*, and *parsimonious*, as a minimal number of informative points is selected, and *general-purpose* since it is based on properties of Gaussian processes like uncertainty and gradients that can be obtained for any differentiable covariance function. This paper builds on of the preliminary work in [26], extending it in several directions. A general framework is provided before describing some specific implementations, extending the study proposing the use of a range of different acquisition functions. A theoretical demonstration of the utility of a gradient term in active sampling and emulation is also given (see Appendix A, for instance). Finally, more thorough experimental results are provided, with more examples and challenging model comparisons, and a more advanced use case.

1.4. Structure of the paper

The remainder of the paper is organized as follows. We first define active emulation and establish the notation in Section 2. Then, the GP-based active emulation framework is presented in Section 3. The framework defines a general-purpose acquisition function built on optimal search of diversity and uncertainty criteria. Experimental results in synthetic and challenging real problems illustrate the capabilities in Section 4. We will pay special attention to the field of remote sensing, where computer codes, called radiative transfer models (RTMs), are widely used and pose challenges to the design of accurate and compact emulators. Having access to an exhaustive ground truth allows us to analyze performance in terms of convergence and accuracy. We conclude in Section 5 with some remarks and an outline of future work.

2. Active emulation

In this section we describe the generic active emulation (AE) method for a complex system denoted as $\mathbf{f}(\mathbf{x})$, e.g., an expensive RTM model. We first fix the notation, then present the processing scheme. Consider a D-dimensional bounded input space $\mathcal{X}$, i.e., $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$. Furthermore, let $\mathbf{f}(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^P$ denote a complex system with P outputs. Finally, $t \in \mathbb{N}$ denotes the index of the AE algorithm, and m_t the number of datapoints $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{m_t}$ used by the algorithm at iteration t, where

$$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k), \tag{1}$$

where $\mathbf{y}_k = [y_{1,k}, \dots, y_{P,k}]^\intercal$ and $k = 1, \dots, m_t$. Thus, given an input matrix of nodes, $\mathbf{X}_t = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_t}]$ of dimension $D \times m_t$, we have a $P \times m_t$ matrix of outputs, $\mathbf{Y}_t = [\mathbf{y}_1, \dots, \mathbf{y}_{m_t}]$. At each iteration t, given the datapoints $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{m_t}$, the AE method constructs an interpolating function $\widehat{\mathbf{f}}_t(\mathbf{x})$. Then, an acquisition function $A_t(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$ is built in order to suggest which regions of the space require additional nodes. That is, an optimization step is performed for obtaining the next input $\mathbf{x}_{m_t+1}$:

$$\mathbf{x}_{m_t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} A_t(\mathbf{x}). \tag{2}$$

The dataset is updated accordingly, $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_{m_t+1}]$, $\mathbf{Y}_{t+1} = [\mathbf{Y}_t, \mathbf{y}_{m_t+1} = \mathbf{f}(\mathbf{x}_{m_t+1})]$ adding a new node, and we set $m_{t+1} = m_t + 1$ and $t \leftarrow t + 1$. The procedure is repeated until a stopping condition is met. One possibility is to stop the algorithm when a pre-established maximum number of points M (determined by the available computational resources) has been included. Theoretically, the user could stop the algorithm when a least a precision

[1] In the following, the words *node* and *datapoint* will be used interchangeably.

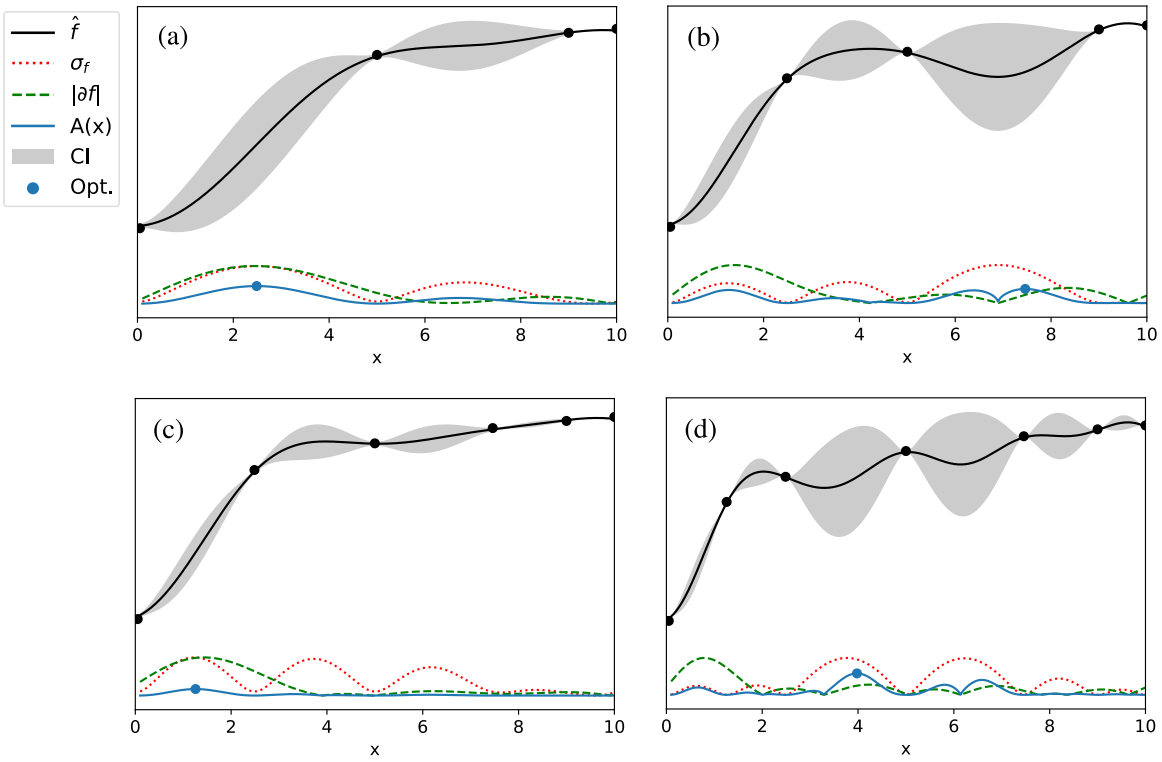


Fig. 1. The presented method optimizes the selection of most informative points to approximate an arbitrary multidimensional function iteratively. The example shows the first four iterations in a 1D case. Starting from 4 points, a GP interpolator is built from which some valuable information is derived (the predictive variance -green- and the gradient -red-) and then combined in an acquisition function (blue) that proposes the next point to sample (blue dot). The acquisition function admits many general forms and trades off geometry and diversity terms to account for attractiveness in the sample space. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Main notation of the work.

$t \in \mathbb{N}$	Iteration index of the active emulator.
m_t	Number of data points at the tth iteration.
$\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{m_t}$	Data points at the tth iteration.
$\mathbf{x} = [x_1, \ldots, x_D]^\intercal \in \mathcal{X} \subset \mathbb{R}^D$	Input variable.
$\mathbf{y} = [y_1, \ldots, y_P]^\intercal$	Outputs.
$\mathbf{X}_t = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_t}]$	$D \times m_t$ input matrix.
$\mathbf{Y}_t = [\mathbf{y}_1, \ldots, \mathbf{y}_{m_t}]$	$P \times m_t$ output matrix.
$\mathbf{y} = \boldsymbol{f}(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^P$	Unknown function/forward model linking $\mathbf{x}$ with $\mathbf{y}$.
$\widehat{\mathbf{y}} = \widehat{\mathbf{f}}_t(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^P$	Interpolator the tth iteration using $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{m_t}$.
$A_t(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$	Acquisition function at the tth iteration.
$k(\mathbf{x}, \mathbf{z}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	kernel function.
$\mathbf{K}$	$m_t \times m_t$ kernel matrix.
$\mathbf{k}_x = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_{m_t})]^\intercal$	$m_t \times 1$ vector.

error $\epsilon > 0$ is achieved, $\|\mathbf{f}(\mathbf{x}) - \widehat{\mathbf{f}}_t(\mathbf{x})\| \leq \epsilon$. However, since $\mathbf{f}(\mathbf{x})$ is costly and analytically intractable in general[2], one cannot evaluate and/or approximate the associated error $\|\mathbf{f}(\mathbf{x}) - \widehat{\mathbf{f}}_t(\mathbf{x})\|$. A practical alternative is to stop the AE method when $\|\widehat{\mathbf{f}}_t(\mathbf{x}) - \widehat{\mathbf{f}}_{t-1}(\mathbf{x})\| \leq \epsilon'$ for some $\epsilon' > 0$. Fig. 1 shows a graphical representation of a generic AE procedure. Table 1 summarizes the main notation of the work. Table 2 shows in details the steps of a generic AE algorithm. Note that the goal is either to sequentially construct an emulator able to obtain a pre-established error in approximation with the smallest number of nodes possible or, more commonly, the best possible emulator built with a pre-established maximum number of nodes (given some starting points). The constructed emulator will be used for further applications for the interested users, researchers and practitioners. We do not consider time or computational restrictions in the construction stage. Furthermore, our approach is particularly useful when the underlying function is very costly, i.e. when the cost of evaluating this function is significantly greater than the application of one iteration of the proposed algorithm.

[2] The system $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is a black-box mapping, linking the inputs $\mathbf{x}$ with the outputs $\mathbf{y}$. At each new input $\mathbf{x}'$, the system returns $\mathbf{y}' = \mathbf{f}(\mathbf{x}')$, but does so by way of a computer code which is too complex and slow to lend itself to exhaustive analysis across the input space.

2.1. Acquisition function

We consider acquisition functions $A_t(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$ obtained by the multiplication of a *geometry* term $G_t(\mathbf{x})$ and a *diversity* factor $D_t(\mathbf{x})$, i.e. functions of the form:

$$A_t(\mathbf{x}) = [G_t(\mathbf{x})]^{\beta_t} D_t(\mathbf{x}), \tag{3}$$

where $\beta_t \in [0, 1]$ is a positive non-decreasing function of t, with $\lim_{t \to \infty} \beta_t = 1$. The function $G_t(\mathbf{x})$ encodes the geometrical information in $\mathbf{f}(\mathbf{x})$, while function $D_t(\mathbf{x})$ depends on the distribution of the points in the current vector $\mathbf{X}_t$. More specifically, $D_t(\mathbf{x})$ takes

Table 2
Generic active emulator.

1. Set $t = 0$, select initial points $\mathbf{X}_0 = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_0}]$, and $\mathbf{Y}_0 = [\mathbf{y}_1, \ldots, \mathbf{y}_{m_0}]$, and maximum number of nodes M.
2. While $m_t < M$:
(a) Given $\mathbf{X}_t = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_t}]$ and $\mathbf{Y}_t = [\mathbf{y}_1, \ldots, \mathbf{y}_{m_t}]$, build function $\widehat{\mathbf{f}}_t(\mathbf{x})$.
(b) Build the acquisition function $A_t(\mathbf{x})$ from $\widehat{\mathbf{f}}_t$, and obtain the new input
$\mathbf{x}_{m_t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} A_t(\mathbf{x})$. (5)
(c) Obtain outputs $\mathbf{y}_{m_t+1} = \mathbf{f}(\mathbf{x}_{m_t+1})$.
(d)Update $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_{m_t+1}]$, $\mathbf{Y}_{t+1} = [\mathbf{Y}_t, \mathbf{y}_{m_t+1}]$.
(e) Set $m_{t+1} = m_t + 1$ and $t \leftarrow t + 1$.
3. Build the interpolating function $\widehat{\mathbf{f}}_t(\mathbf{x})$.
4. Return final set of optimal nodes $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{m_t}$ as a Look-up Table (LUT), as well as the gradient and the predictive variance of the predictive model $\widehat{\mathbf{f}}_t(\mathbf{x})$.

greater values around empty areas in $\mathcal{X}$, whereas $D_t(\mathbf{x})$ will be approximately zero close to the nodes and exactly zero at the nodes[3], i.e., $D_t(\mathbf{x}_i) = 0$, for $i = 1, \ldots, m_t$ and $\forall t \in \mathbb{N}$. As a consequence, we have

$$A_t(\mathbf{x}_i) = 0 \quad \forall i, t. \tag{4}$$

Generally, since $\mathbf{f}(\mathbf{x})$ is analytically intractable, the function $G_t(\mathbf{x})$ can only be derived from information acquired in advance or by considering the approximation $\widehat{\mathbf{f}}_t(\mathbf{x})$. The *tempering parameter*, β_t, helps to down-weight the likely less informative estimates of the gradient in the very first iterations. For instance, if $\beta_t = 0$, we ignore $G_t(\mathbf{x})$ and $A_t(\mathbf{x}) = D_t(\mathbf{x})$, i.e., only the exploration term is considered. Whereas, if $\beta_t = 1$, we have $A_t(\mathbf{x}) = G_t(\mathbf{x})D_t(\mathbf{x})$.

2.2. Specific implementation

The AE algorithm introduced is completely defined by the choice of the interpolator providing the approximation $\widehat{\mathbf{f}}_t(\mathbf{x})$, and the functions $G_t(\mathbf{x})$, $D_t(\mathbf{x})$, and β_t. Moreover, the initial set of nodes $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{m_0}$ and the stopping condition could be considered as additional elements. It is important to note that, in order to choose the interpolating function, we have to take into account the ease of application in high dimensional spaces and the possibility of computing the gradient and other differential geometric measures of $\widehat{\mathbf{f}}_t$ analytically. Different designs of these four elements give rise to different AE techniques. In Section 3, we provide some specific examples of the choice of $\{\widehat{\mathbf{f}}_t, G_t, D_t, \beta_t\}$.

2.3. Parsimonious sequential approach

It is also important to remark that the active emulation procedure presented in this work is intrinsically a sequential technique. This means that the nodes in $\mathbf{X}_{t-1}$ are always contained in $\mathbf{X}_t$, i.e. the locations of previous nodes are not changed. This solution minimizes the number of evaluations of the complex system $\mathbf{f}$. In this sense, the active emulation procedure is a parsimonious sequential technique that applies, at each iteration, all previously obtained information about the underlying function $\mathbf{f}$. Namely, all the previous evaluations of $\mathbf{f}$ are used, and only one additional evaluation of $\mathbf{f}$ is required at each iteration.

2.4. Products of the algorithm

The active emulation procedure proposed in this work is a methodology that delivers: (a) an accurate GP emulator (considering a specific choice of the interpolator) while evaluating the computer code as little as possible, (b) a final set of nodes $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{m_t}$ as a Lookup Table (LUT; other interpolation procedures can be applied using the obtained set of points), and (c) useful statistical information about the model $\mathbf{f}$, such as predictive variance and gradients of the learned function, which can be further used for model inversion and error propagation analyses.

[3] Note that this is the case only for an interpolator (no output-noise assumed) while for a regressor the value of $D_t(\mathbf{x})$ will just be very small around already placed nodes.

3. Active multi-output Gaussian process emulator (AMOGAPE)

An active emulator is completely defined by the choice of the predictive, model $\widehat{\mathbf{f}}(\mathbf{x})$ and the acquisition function $A_t(\mathbf{x})$. In this work, we consider a GP interpolator, as well as the regression formulation [27], which has been successfully used in remote sensing applications recently [28].

3.1. The Gaussian process interpolator

For the sake of simplicity, let us first start considering the GP solution for the scalar output case, i.e., $P = 1$. Hence, in this case the vectorial function $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is a simple function $y = f(\mathbf{x})$, and the matrix $\mathbf{Y}_t = [y_{1,1}, \ldots, y_{1,m_t}]$, becomes a $1 \times m_t$ vector. Given a generic test input $\mathbf{x}$, GPs provide a Gaussian predictive density $p(y|\mathbf{x}) = \mathcal{N}(y|\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$ with predictive mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$. The predictive mean gives us the interpolating function and is given by

$$\widehat{f}_t(\mathbf{x}) = \mu_t(\mathbf{x}) = \mathbf{k}_x^\intercal \mathbf{K}^{-1} \mathbf{Y}_t^\intercal, \tag{6}$$

where we defined a kernel function $k(\mathbf{x}, \mathbf{z}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the corresponding kernel matrix $[\mathbf{K}]_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ of dimension $m_t \times m_t$ containing all kernel entries, and the kernel vector $\mathbf{k}_x = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_{m_t})]^\intercal$ of dimension $m_t \times 1$. The interpolating function can be simply expressed as a linear combination of $\hat{f}_t(\mathbf{x}) = \mathbf{k}_x^\intercal \boldsymbol{\alpha} = \sum_{i=1}^{m_t} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$, where the weights $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{m_t}]^\intercal$ are $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{Y}_t^\intercal$. The GP formulation also provides an expression for the predictive variance

$$\sigma_t^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_x^\intercal \mathbf{K}^{-1} \mathbf{k}_x. \tag{7}$$

An example is the exponentiated quadratic kernel function,

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\delta^2}\right), \tag{8}$$

where $\|\cdot\|$ is the ℓ_2-norm, and $\delta > 0$ is a positive scalar hyperparameter. Note that the norm of the gradient of the interpolating function $\widehat{f}_t$ w.r.t. the input data $\mathbf{x}$ can be easily computed,

$$\mathtt{Gr}_t(\mathbf{x}) = \left\| \nabla_x \widehat{f}_t(\mathbf{x}) \right\| = \left\| \sum_{i=1}^{m_t} \alpha_i \nabla_x k(\mathbf{x}, \mathbf{x}_i) \right\|. \tag{9}$$

The gradient vector of $k(\mathbf{x}, \mathbf{x}_i)$ with $\mathbf{x} = [x_1, \ldots, x_D]^\intercal$ and $\mathbf{x}_i = [x_{1,i}, \ldots, x_{D,i}]^\intercal$, is

$$\nabla_x k(\mathbf{x}, \mathbf{x}_i) = -\frac{k(\mathbf{x}, \mathbf{x}_i)}{\delta^2}[(x_1 - x_{1,i}), \ldots, (x_D - x_{D,i})]^\intercal, \tag{10}$$

which can be easily computed analytically, and by automatic differentiation software. At this point an intuitive choice of acquisition function of Eq. (3) presents itself. The predictive variance which describes the uncertainty of the GP prediction, and which largely depends on distance to nearby training points, is a natural choice for the diversity term $D_t(\mathbf{x}) = \sigma_t^2(\mathbf{x})$. Furthermore, since the emulator is differentiable, we can use the gradient as a measure of function variation and choose the geometry term as $G_t(\mathbf{x}) = \mathtt{Gr}_t(\mathbf{x})$.

3.2. Multi-output GP interpolator

Several multi-output GP schemes have been proposed with the aim of exploiting the correlation among the output variables [29–34]. These models are especially well suited for multitask problems where little data is available or for gap filling, which is not the scenario of this work [30]. We do not face such problems in our particular remote sensing application since the RTMs provide all vector components when executed in forward mode. We adopt a simpler yet highly effective approach, simply treating each output independently. For simplicity, we consider an isotopic case where to each input $\mathbf{x}_k$ we have P different outputs, $[y_{1,k}, \dots y_{P,k}]^\intercal$; see the descrption of isotopic and heterotopic models in [30]. We also define the pth row of the matrix $\mathbf{Y}_t$ as $\widetilde{\mathbf{y}}_{p,t} = [y_{p,1}, \dots, y_{p,m_t}]$, with $p = 1, \dots, P$, so that $\mathbf{Y}_t$ is matrix of dimension $P \times m_t$. Here, for the sake of simplicity, we apply one GP interpolator for each output independently, i.e.,

$$\widehat{\mathbf{f}}_t(\mathbf{x}) = \begin{cases} \widehat{f}_{1,t}(\mathbf{x}) = \mathbf{k}_{x,1}^\intercal \mathbf{K}_1^{-1} \widetilde{\mathbf{y}}_{1,t}^\intercal \\ \quad \vdots \qquad\qquad , \\ \widehat{f}_{P,t}(\mathbf{x}) = \mathbf{k}_{x,P}^\intercal \mathbf{K}_P^{-1} \widetilde{\mathbf{y}}_{P,t}^\intercal \end{cases} \tag{11}$$

where the vectors $\mathbf{k}_{x,p}$ have all dimension $m_t \times 1$ and the matrices $\mathbf{K}_p$ have dimension $m_t \times m_t$. The subindex p in the kernel vector $\mathbf{k}_{x,p}$ and the kernel matrix $\mathbf{K}_p$ denotes the dependence to a different hyper-parameter δ_p (we learn one for each output). More generally, we can consider a different kernel for each output which allows for much model flexibility. Hence, for each output, we have a different variance

$$\sigma_{p,t}^2(\mathbf{x}) = k_p(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{x,p}^\intercal \mathbf{K}_p^{-1} \mathbf{k}_{x,p}. \tag{12}$$

Similarly, we have one gradient norm for each interpolating function $\mathtt{Gr}_{p,t}(\mathbf{x})$. It is important to note here that any multi-output GP framework will fit in the AMOGAPE method as long as it provides a differentiable predictive variance and gradient, which for most commonly used kernels is the case.

3.3. The acquisition function

Note that $\sigma_p^2(\mathbf{x}_i) = 0$ for all $i = 1, \dots, m_t$ and all p, and each $\sigma_p^2(\mathbf{x})$ depends on the distance among the support points $\mathbf{x}_t$, the chosen kernel function k, and the value of the corresponding hyper-parameter δ_p. For this reason, it is reasonable to consider as diversity term the following function that combines them all:

$$D_t(\mathbf{x}) := \sigma_{1,t}^2(\mathbf{x}) \odot \sigma_{2,t}^2(\mathbf{x}) \odot \sigma_{3,t}^2(\mathbf{x}) \dots \odot \sigma_{P,t}^2(\mathbf{x}), \tag{13}$$

where $\odot$ represents a generic mathematical operation such as sum (+) or multiplication ($\times$). We wish to use the geometric information term to sample where the norm of the gradient is high and thus define similarly

$$G_t(\mathbf{x}) := \mathtt{Gr}_{1,t}(\mathbf{x}) \odot \mathtt{Gr}_{2,t}(\mathbf{x}) \odot \mathtt{Gr}_{3,t}(\mathbf{x}) \dots \odot \mathtt{Gr}_{P,t}(\mathbf{x}). \tag{14}$$

The intuition behind this choice is that wavy regions of $\mathbf{f}$ (estimated by $\widehat{\mathbf{f}}_t$) require more support points than flat regions. In Appendix A we demonstrate the importance of the gradient term using the simple example of a piecewise-constant interpolator. As previously mentioned, we define the acquisition function as

$$A_t(\mathbf{x}) = [G_t(\mathbf{x})]^{\beta_t} D_t(\mathbf{x}). \tag{15}$$

Table 3 shows several combinations that generate different acquisition functions according to the choice of the operator $\odot$.

3.3.1. Optimization approaches

The maximization of Eq. (15) can be performed by using different optimization algorithms, e.g., gradient ascent or simulated annealing. As can be seen in the simple 1-D example of Fig. 1, the acquisition function has many local optima. Thus, while it is useful to have access to the gradient of Eq. 15, we find that it is important to incorporate stochasticity in the optimization. This can be done, for example by performing a number of random searches and then performing gradient ascent, initialized at the best candidate point.

3.3.2. Tempering of the geometric information

The parameter $\beta_t \in [0, 1]$ indicates how the acquisition function should "trust" the provided geometric information and must be an non-decreasing function of t. Indeed, recall that the geometric information is given by analyzing the interpolating function $\widehat{\mathbf{f}}$, instead of the complex system $\mathbf{f}$, since it is analytically intractable. Clearly, β_t must be an increasing function with respect to t, since at each iteration the interpolating function $\widehat{\mathbf{f}}$ is improved and becomes step-by-step more reliable. One possible choice is $\beta_t = 1 - \exp(-\gamma t)$, where $\gamma \geq 0$ is a positive scalar established by the user or, alternatively, $\beta_t = 1 - \frac{1}{t}$, for instance.

3.4. From interpolation to regression

So far we have described the emulation as an interpolation problem since RTMs are deterministic models: running the code multiple times will always return identical answers. Hence, we have assumed an observation equation of type $y = f(\mathbf{x})$[4] However, in some cases, it is preferable to consider an observation equation of type $y = f(\mathbf{x}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \upsilon^2)$ represents a Gaussian noise perturbation with zero mean and variance υ^2. There are three main reasons, both theoretical and practical, for considering noisy outputs: (a) the system to emulate actually contains stochastic elements (i.e., it is not a completely deterministic system), (b) to increase the prediction power of the emulator function $\widehat{f}_t(\mathbf{x})$ providing more flexibility to the GP model, and (c) in order to avoid numerical problems, increasing also the stability of the computation. This last point is due to the fact that the noise variance ν^2 plays the role of a regularization term which is added to the diagonal of the kernel matrix (also called a *nugget* in kriging literature). Indeed, when noisy outputs are assumed and by denoting the $m_t \times m_t$ identity matrix as $\mathbf{I}$, then the GP regression equations become

$$\widehat{f}_t(\mathbf{x}) = \mathbf{k}_x^\intercal (\mathbf{K} + \upsilon^2 \mathbf{I})^{-1} \mathbf{Y}_t^\intercal, \tag{16}$$

$$\sigma_t^2(\mathbf{x}) = \upsilon^2 + k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_x^\intercal (\mathbf{K} + \upsilon^2 \mathbf{I})^{-1} \mathbf{k}_x. \tag{17}$$

Note that, if we set again $D_t(x) = \sigma_t^2(\mathbf{x})$ (with $\sigma_t^2(\mathbf{x})$ defined above), then $A_t(\mathbf{x})$ does not fulfil Eq. (4). However, $A_t(x)$ still takes greater values far from nodes $\mathbf{x}_i$, and smaller values close to points $\mathbf{x}_i$. If the application strictly requires that the condition in Eq. (4) must be satisfied, then we can simply define $D_t(x) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_x^\intercal \mathbf{K}^{-1} \mathbf{k}_x$, i.e., $\sigma^2(\mathbf{x})$ without the noise term. With this definition, we have again $A_t(\mathbf{x}_i) = 0$. This means that the noise

[4] In this section, we assumed only one output in the equation, just for the sake of simplicity. Clearly, the same considerations are valid for the multi-output case.

term is only used in the GP equations and not for the construction of the acquisition function. Finally, note that, if a regressor is applied instead of an interpolator, then two hyperparameters must be tuned, δ and υ, instead of just only δ, assuming the kernel in Eq. (8). The user might also wish to decide a value of υ^2 in advance instead of learning it, using it as a regularization term in order to guarantee the numerical stability of the method. Hyperparameter tuning can be performed with standard Cross Validation (CV) procedures, or maximizing the marginal likelihood function by gradient ascent or other optimization techniques [35,36]. In the interpolation case or when υ^2 is decided in advance by the user, another interesting approach is to find the maximum value of bandwidth δ which still allows the numeric inversion of the matrix $\mathbf{K}$ (imposing a upper bound for its condition number).

4. Experimental results

This section presents experimental results of the our AE framework in synthetic and real (Earth-observation) systems. The AMOGAPE[5] method is compared to standard algorithms in the literature, namely random exploration/sampling and most notably Sobol's sampling [37] and the Latin Hypercube Sampling (LHS) method [38]. Algorithms are compared in terms of accuracy and convergence rates in problems of different input and output dimensionality. The real experiments involve a widely used code that models the relation between vegetation parameters and the corresponding reflectance signal.

4.1. Toy experiment 1: example of unidimensional multi-output emulation

We consider a multi-output toy example with scalar inputs $x \in \mathbb{R}$ where we can easily compare the achieved approximation $\widehat{\mathbf{f}}_t(x) = [\widehat{f}_1(x), \widehat{f}_2(x)]$ with the underlying function $\mathbf{f}(x) = [f_1(x), f_2(x)]$. In this way, we can exactly check the true accuracy of the obtained approximation using different schemes. For the sake of simplicity, we consider the following multi-output mapping

$$\mathbf{f}(x) = [\log(x), 0.5\log(3x)], \quad x \in (0, 10], \tag{18}$$

then $D = 1$ and $P = 2$ (two outputs). Even in this simple scenario, the procedure used for selecting new points is relevant. We start with $m_0 = 4$ support points, $\mathbf{X}_0 = [0.1, 3.4, 6.7, 10]$, apply an independent GP per output, and for AMOGAPE we use the acquisition function denoted as $\Pi D \times \Pi G$ in Table 3 with the tempering function $\beta_t = 1 - \frac{1}{t}$. We also set $\upsilon^2 = 0.02$ as a regularization term, in order to avoid numerical issues.

4.1.1. Comparison among sequential methods

It is important to remark that all the active emulators presented in this work are intrinsically sequential techniques. This means that the nodes in $\mathbf{X}_{t-1}$ are always contained in $\mathbf{X}_t$, i.e., the previous configuration of points is always kept. Therefore, for a fair comparison we have to consider other sequential algorithms. We add to $\mathbf{X}_t$ sequentially 20 additional points, using different sampling strategies: AMOGAPE, uniform points randomly generated in (0,10], a sequential Sobol sequence, and a sequential version of the Latin Hypercube Sampling procedure (Seq-LHS). Seq-LHS simply generates 20 nodes following the LHS procedure and then adds one to $\mathbf{X}_t$ at each iteration (without replacement). Note that, at each run, the results can vary even for the deterministic procedure due to the optimization of the hyperparameters. We use simulated annealing, which is a stochastic optimization technique [35,36], both for hyperparameter and acquisition function optimization. We average all the results over 500 independent runs. For model comparison, we compute the root mean square error (RMSE) between $\widehat{\mathbf{f}}_t(x)$ and $\mathbf{f}(x)$ at each iteration, and show the evolution of the (averaged) RMSE versus the number of support points m_t (that is $m_t = t + m_0$) in Fig. 2. We can observe that the AMOGAPE scheme outperforms the other methods, providing the smallest RMSEs between $\mathbf{f}(x)$ and $\widehat{\mathbf{f}}_t(x)$.

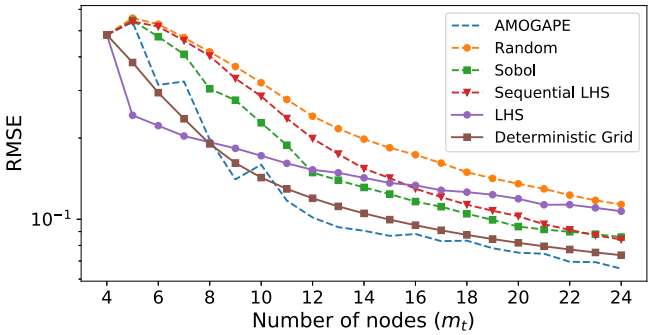


Fig. 2. RMSE (in log-scale) between $\mathbf{f}(x)$ and $\widehat{\mathbf{f}}_t(x)$ versus the number of nodes m_t, that is $m_t = t + 4$ in this example ($D = 1$ and $P = 2$). Sequential methods, which are more comparable as they utilize m_T evaluations of $\mathbf{f}(x)$, are shown with dashed lines. The comparison with two non-sequential methods, using $\sum_{t=1}^{T} m_t = \frac{m_T^2 + m_T}{2}$ evaluations of $\mathbf{f}(x)$, are shown with solid lines.

4.1.2. Comparison with non-sequential methods

In order to provide an exhaustive numerical analysis we also compare AMOGAPE with non-sequential techniques where the input matrix $\mathbf{X}_t$ can be completely different from $\mathbf{X}_{t-1}$ (whereas, in AMOGAPE, the nodes in $\mathbf{X}_{t-1}$ are all always contained in $\mathbf{X}_t$). This approach would not be used in practice, but serves as an interesting comparison of AMOGAPE with one-shot space-filling algorithms. More specifically, we consider:

- Deterministic grid: at each step, we consider an equal-spaced set of points (deterministically chosen). Thus, at each step, all the points in the previous timestep $\mathbf{X}_{t-1}$ are not considered but replaced by new nodes.
- Standard LHS: also in this case, at each iteration all the previous points are changed.

Clearly, these two schemes evaluate the underlying function in m_t new nodes at each iteration and are therefore more costly than AMOGAPE. The total number of evaluations of $\mathbf{f}(x)$ for AMOGAPE is m_T whereas, for the non-sequential schemes above is $\sum_{t=1}^{T} m_t = (m_T^2 + m_T)/2$. However, even in this unfair comparison for our method, Fig. 2 shows that AMOGAPE is able to provide the smallest error when more than 12 new points are incorporated. This illustrates that the gradient term encoded in the AMOGAPE adds useful information to the active learning scheme.

4.2. Toy experiment 2: example of bidimensional multi-output emulation

In this section, we extend the previous example to consider multi-input and multi-output problems, i.e. $D = P = 2$. More specifically, we consider

$$\mathbf{f}(\mathbf{x}) = [\log(|\mathbf{x}|), 0.5\log(3|\mathbf{x}|)], \quad \mathbf{x} \in (0, 10] \times (0, 10]. \tag{19}$$

We start with $m_0 = 25$ starting nodes in the input matrix, $\mathbf{X}_0 = [\mathbf{x}_1, \ldots, \mathbf{x}_{m_0=25}]$ where $\mathbf{x}_i = [x_{i,1}, x_{i,2}]^\intercal$, with $i = 1, \ldots, 25$, distributed as shown in Fig. 3(a) with black circles. In order to evaluate the approximation RMSE obtained with the emulators, we consider a thin grid in the square $(0, 10] \times (0, 10]$ (with step 0.3). The starting nodes in the input matrix, $\mathbf{X}_0$ (black points), and the

[5] Code available at https://github.com/dhsvendsen/AMOGAPE.

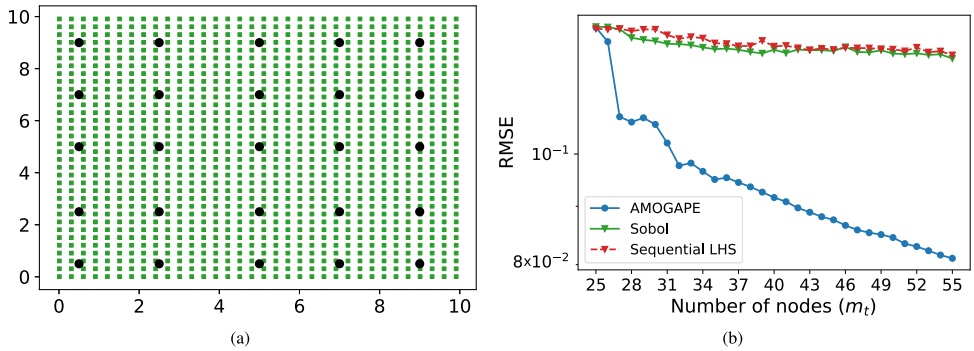


Fig. 3. **(a)** The starting points in the input matrix $\mathbf{X}_0$ are shown with black points, whereas the points in the thin test grid are depicted with green squares. **(b)** RMSE (in log-scale) between $\mathbf{f}(\mathbf{x})$ and $\widehat{\mathbf{f}}_t(\mathbf{x})$ versus the number of the number of support points m_t, that is $m_t = \dots$. in this example ($D = 2$ and $P = 2$). Note that the number of initial points is $m_0 = 25$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

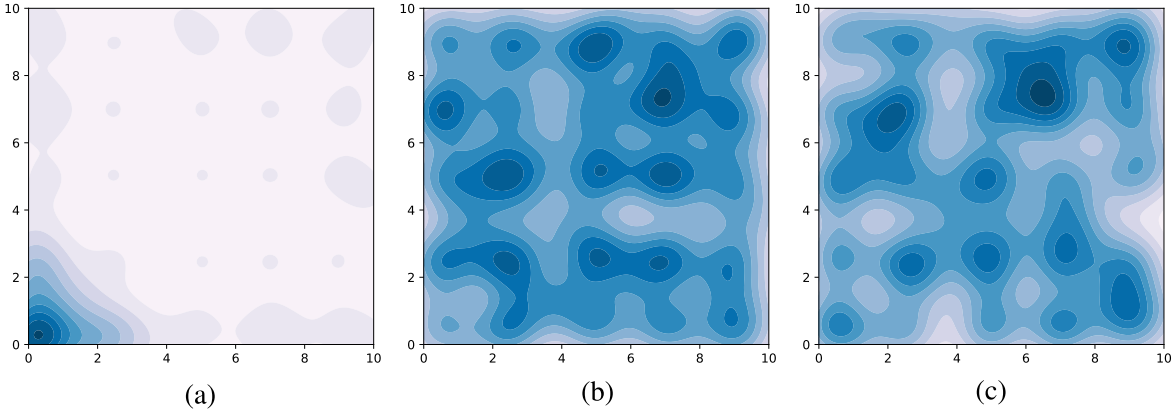


Fig. 4. Kernel Density Estimation plot of the final configuration of points in the emulation of Eq. (19), showing the **(a)** AMOGAPE, **(b)** Sobol and **(c)** Sequential LHS algorithms respectively.

thin test grid (green dots) are shown in Fig. 3(a). We apply again one independent GP for each output and, as in the previous example. For AMOGAPE, we apply the acquisition function denoted as $\Pi D \times \Pi G$ in Table 3 and we use again the tempering function $\beta_t = 1 - \frac{1}{t}$ and set $\upsilon^2 = 0.02$ as a regularization term, only for avoiding numerical issues. We compare different sampling strategies : AMOGAPE, a sequential Sobol sequence, and sequential LHS. We add 30 additional points to $\mathbf{X}_0$ in the first two sequential approaches. In LHS all the previous points change at each iteration. The results (averaged over 500 independent runs) are shown in Fig. 3(b), which show a considerable gain in accuracy and convergence rates by the presented algorithms.

The distributions in input-space of the final 55 nodes - 25 on a grid and 30 subsequently chosen with a sampling algorithm - are shown as a Kernel Density Estimation (KDE) plot for each of the methods in Fig. 4. We can observe that AMOGAPE adds points in the border and in a left-bottom corner where the gradient is comparatively high. It makes sense that these points are deemed the most useful since the initial 25 nodes points are well-located. The sampling method using the Sobol algorithm and the sequential LHS algorithms incorporate new points that fill out the input-space but do not pay particular attention to the behaviour of the underlying function.

4.3. Application to remote sensing: Emulating a radiative transfer model

Our method is assessed for the emulation of the leaf-canopy PROSAIL RTM, which is the most widely used RTM over the last two decades in remote sensing studies [39]. PROSAIL simulates reflectance as a function of:

1. *Leaf optical properties*, given by the mesophyll structural parameter (N), leaf chlorophyll (Chl), dry matter (Cm), water (Cw), carotenoid (Car) and brown pigment (Cbr) contents.
2. *Canopy level characteristics*, determined by leaf area index (LAI), the average leaf angle inclination (ALA) and the hot-spot parameter (Hotspot). System geometry is described by the solar zenith angle (θ_s), view zenith angle (θ_v), and the relative azimuth angle between both angles ($\Delta\Theta$).

We consider PROSAIL for simulating Landsat-8 spectra, a satellite sensor widely used for land cover applications in general and vegetation monitoring in particular. Therefore, the generated, eventually optimized, look-up tables are used for inversion and thus retrieve vegetation parameters with the Landsat-8 satellite imagery. This leaves us with an output-dimension of $P = 9$ for our problem, i.e. the number of spectral bands of the satellite. Now, depending on the parameters of interest the input dimensionality D may vary.

4.3.1. Sampling a 2-dimensional space for PROSAIL emulation

In this experiment, we chose the most important variables at leaf and canopy-level respectively, namely Chl and LAI, and kept the rest fixed. Table 4 shows the values for the remaining parameters which are set for simulation of rice crops [40]. When generating look-up tables with RTMs it is common practice to use expert knowledge to determine distributions over the biophysical parameters which constitute the RTM input [41]. The desired amount of samples are then drawn, and the model is evaluated in each of these points. A commonly used distribution is the truncated Gaussian $\mathcal{N}_{\mathcal{T}}(\mathbf{x}|\mu, \sigma, \min, \max)$. Indeed, the truncated Gaussians $\mathcal{N}_{\mathcal{T}}(\text{Chl}|45, 30, 20, 90)$ and $\mathcal{N}_{\mathcal{T}}(\text{LAI}|3.5, 4.5, 0, 10)$ for Chl and LAI, respectively, have proven effective for crop reflectance model-

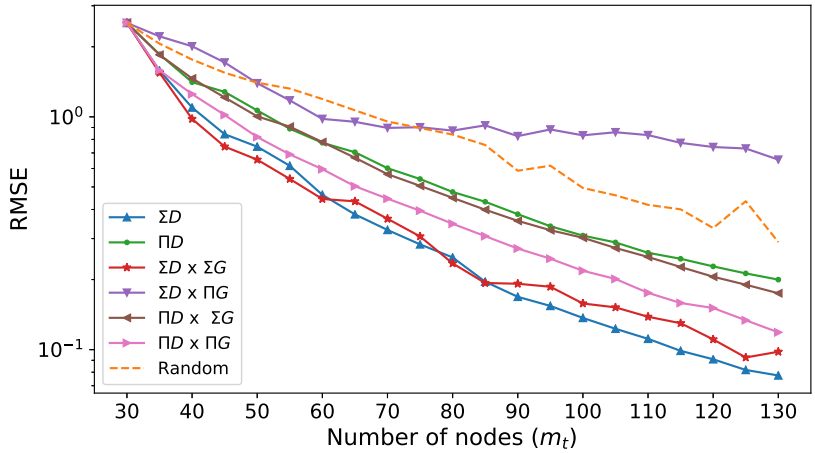


Fig. 5. Function approximation errors by different acquisition functions, cf. Table 3, and for different number of selected nodes m_t in a bidimensional PROSAIL problem. Only the best performing acquisition functions are compared here to random sampling.

Table 3
Acquisition functions for a multi-output emulator and their shorthand notation used in Section 4.

$A_t(\mathbf{x})$	Shorthand
$\sum_{p=1}^{P} \sigma_{p,t}^2(\mathbf{x})$	ΣD
$\prod_{p=1}^{P} \sigma_{p,t}^2(\mathbf{x})$	ΠD
$\sum_{p=1}^{P} \sigma_{p,t}^2(\mathbf{x}) \sum_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΣD × ΣG
$\sum_{p=1}^{P} \sigma_{p,t}^2(\mathbf{x}) \prod_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΣD × ΠG
$\prod_{p=1}^{P} \sigma_{p,t}^2(\mathbf{x}) \sum_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΠD × ΣG
$\prod_{p=1}^{P} \sigma_{p,t}^2(\mathbf{x}) \prod_{p=1}^{P} \mathtt{Gr}_{p,t}(\mathbf{x})$	ΠD × ΠG

ing [40]. We denote their joint distribution, which has no covariance between the variables, as $\mathcal{N}_{\mathcal{T}}$(Chl,LAI).

In summary, we are emulating a function f($\mathbf{x}$) where $\mathbf{x}$ = [Chl, LAI] mapping from an input space of dimension $D = 2$ to the output space of dimension $P = 9$. The search space is restricted to physically meaningful values of Chl $\in [20; 90]$ µg/cm^2 and LAI $\in [0; 10]$. In order to gain insight into the relative importance of the Diversity and Geometric terms, an array of different acquisition functions shown in Table 3 are applied. The AMOGAPE sampling schemes are compared with sampling randomly from $\mathcal{N}_{\mathcal{T}}$(Chl, LAI). This distribution encodes knowledge about the physically feasible region to sample in, which is also encoded in the AMOGAPE, simply by multiplying the truncated density function ψ(Chl, LAI) onto the acquisition functions. We set $\beta_t = 1\ \forall\ t$ in order to simplify the experiments.

Evaluation of which sampling method leads to the best emulator is done by computing the test approximation error on a test-set of 5000 points, sampled from the above-mentioned truncated Gaussian distributions. We initialize with 30 points drawn from $\mathcal{N}_{\mathcal{T}}$(Chl,LAI). The multi-output RMSE for the $M = 5000$ test points over the $P = 9$ single-output GP emulators is computed as follows

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\frac{1}{P}\sum_{p=1}^{P}(y_{p,i} - \widehat{y}_{p,i})^2}. \tag{20}$$

The results are averaged over 15 runs. In order to speed up the experiment, hyperparameter and acquisition function optimization are performed through an initial random search of 10^D points, followed by gradient ascent. Results are shown in Fig. 5. We see that it is possible to perform better using the AMOGAPE approach on our test-set than by sampling randomly from $\mathcal{N}_{\mathcal{T}}$(Chl,LAI). It

Table 4
Characteristics of the simulation used in the PROSAIL model.

Leaf level	N	Cm	Cw	Car	Cbr
	1.5	0.01 µg/cm^2	0.01 µg/cm^2	8 g/cm^2	0
Canopy level	ALA	Hotspot	θ_s	θ_v	$\Delta\Theta$
	Spherical	0.01	30°	10°	0

is interesting to note that methods using ΣD × ΣG and ΣD perform similarly, implying that the ΣD term is governing the acquisition function. Similarly, methods using ΠD and ΠD × ΣG perform equally well, showing that ΠD is the most influential term. The acquisition function ΣD × ΠG, which penalizes a zero-gradient in any of the output dimension, relies too much on geometric information and performs the worst. It seems that the information source which is included in product form governs $A(\mathbf{x})$. It seems that the ΠD × ΠG method manages to strike a balance between the two sources of information. All in all, the best performing methods are ΣD and ΣD × ΣG. This hints at the idea that the product form is too restrictive, i.e. considering a point uninteresting if the predictive variance is close to zero in only one of the output-dimensions.

4.3.2. Sampling a 3-dimensional space for PROSAIL emulation

We conduct a similar experiment, including now another crucial biophysical parameter in the search space, namely dry matter content (Cm), which is an important parameter to monitor key properties and processes in vegetation and the wider ecosystem. The associated truncated Gaussian used is $\mathcal{N}_{\mathcal{T}}$(Cm|0.005, 0.005, 0.003, 0.011). We use a test set of 50,000 points generated from the joint truncated Gaussian $\mathcal{N}_{\mathcal{T}}$(Chl, LAI, Cm).

We saw earlier that the acquisition function which performs the best was also the most simple, namely ΣD. The acquisition function ΠD × ΠG, being formulated only in product form, manages not to be dominated by either term and is interesting because it is very selective: It discourages a gradient or predictive variance which is close to zero in any output dimension. For these reasons, along with computational burden, the aforementioned acquisition functions are used for the 3-dimensional experiment. The average results after running the experiment 10 times are shown in Fig. 6. Again, we see that the 1) the two variants of AMOGAPE acquisition functions outperform random sampling, 2) that the acquisition functions behave quite similarly, and 3) that simple acquisitions perform as well as more complicated ones. Note however, that

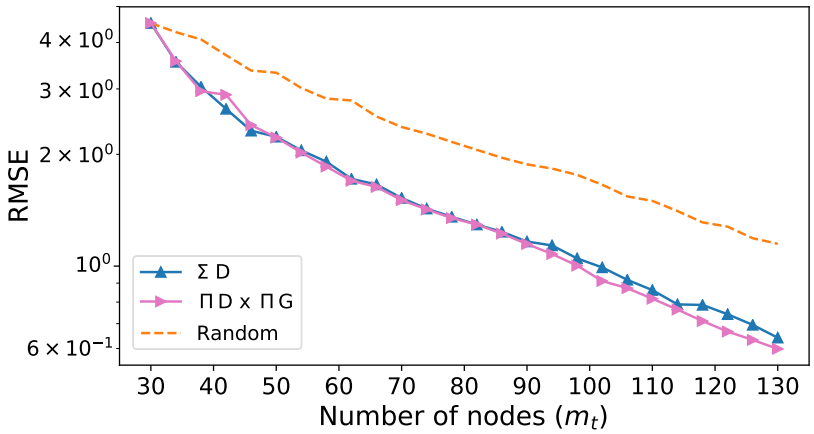


Fig. 6. Function approximation errors by different acquisition functions and for different number of selected nodes m_t in the three-dimensional PROSAIL problem.

using a different tempering function than $\beta_t = 1 \, \forall \, t$ would likely make performances diverge.

5. Conclusions

We introduced a simple framework for active construction of emulators for costly physical models used in Earth observation. The proposed framework does not only provide an effective approximating function, but also a compact LUT and some very useful by-products for practitioners, namely confidence intervals for the estimates and information about the gradients.

The methodology iteratively incorporates new sample points that meet both diversity and geometry criteria, thus sampling in low-density and more 'complex' regions. This is accomplished by building an acquisition function that takes into account the predictive variance and the norm of the gradient of the GP function used for emulation. The combination of the geometric and diversity sampling criteria was possible because both the GP predictive variance and the gradient of the GP predictive mean are analytic expressions.

We illustrated the promising capabilities of the method through emulation of a popular radiative transfer model. Comparison to established methods in the literature illustrated the favourable performance of the proposed methods. The proposed family of criteria for defining the acquisition functions in emulation allows smart sampling of the input space thus leading to compact and expressive look-up-tables, which can be readily used for model inversion in either statistical or numerical frameworks. The proposed methodology is very general and modular. Alternative acquisition functions, kernel functions and quality measures adapted to the problem are interesting pathways to explore. In our future work we plan to explore the use of Matérn kernels when function smoothness is not a strict (or even realistic) requirement in a given RTM. Besides, other quality measures other than RMSE could be more interesting for evaluating emulator quality. The information content of the added samples in each iteration by computing maximum differential of entropies in similar ways to the approach in [42].

We anticipate adoption of these methods in the Earth sciences and also in unrelated disciplines where process-based models are widely adopted as well, from econometrics to industry or health sciences. Our future work is centered around speeding up other more complex codes, such as the atmosphere MODTRAN model, as well as to extend the framework to deal with dynamic models. The framework introduced here constitutes the first step towards the ambitious goal of large scale active statistical models that learn Physics models.

Declaration of Competing Interest

None.

Acknowledgments

This work is supported by the European Research Council (ERC) under the ERC-CoG-2014 SEDAL Consolidator grant (grant agreement 647423).

Appendix A. Importance of a gradient term in the acquisition function

Let us consider the problem of approximating the function $y = f(\mathbf{x})$, $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^L$, where $\mathcal{D} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_L, b_L]$. For the sake of simplicity, we consider one dimensional problems, i.e. $L = 1$, with $\mathcal{D}$ bounded a_i, $b_i < \infty$. Moreover, let us consider a set of nodes $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\} \in \mathcal{D}$ and the corresponding values of the function $y_m = f(\mathbf{x}_m)$. We use the pairs $\{\mathbf{x}_m, y_m\}_{m=1}^M$ to perform an interpolation.

Given the set of nodes $\{\mathbf{x}_m\}_{m=1}^M$, we denote a piecewise constant interpolation (PCI) of the function $f(\mathbf{x})$ as

$$\widehat{y} = \phi(\mathbf{x}|\mathbf{x}_{1:M}) = \phi(\mathbf{x}). \tag{A.1}$$

In order to measure the discrepancy between function and emulator we introduce a cost function $C(f, \phi) = C\big(f(\mathbf{x}), \phi(\mathbf{x})\big) \geq 0$. The equality $C(f, \phi) = 0$ must hold only if $\phi(\mathbf{x}) = f(\mathbf{x})$. Here, we consider the L_p family of cost functions

$$C_p(f, \phi) = ||f(\mathbf{x}) - \phi(\mathbf{x})||_p = \left(\int_{\mathcal{D}} |f(\mathbf{x}) - \phi(\mathbf{x})|^p d\mathbf{x} \right)^{\frac{1}{p}}. \tag{A.2}$$

Note that $C_\infty(f, \phi) = \lim_{p \to \infty} C_p(f, \phi) = \max_{\mathbf{x} \in \mathcal{D}} |f(\mathbf{x}) - \phi(\mathbf{x})|$.

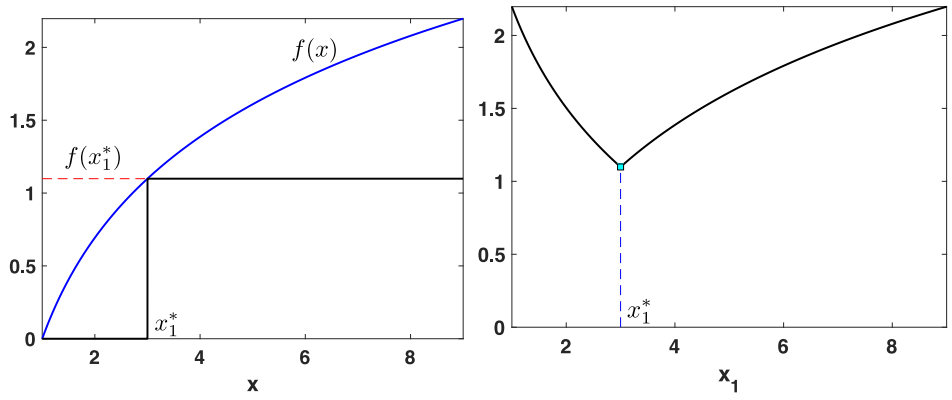
One node ($M = 1$), infinity norm cost functions $p = \infty$

Let us consider $y = f(x)$, $x \in [a, b] \subseteq \mathbb{R}$. For simplicity, we assume $f(x)$ to be strictly monotonic, more specifically increasing. We consider a piecewise constant approximation with $M = 1$ point x_1 within x_1, i.e.

$$\phi(x) = \begin{cases} f(a) & x \leq x_1 \\ f(x_1) & x > x_1 \end{cases} \tag{A.3}$$

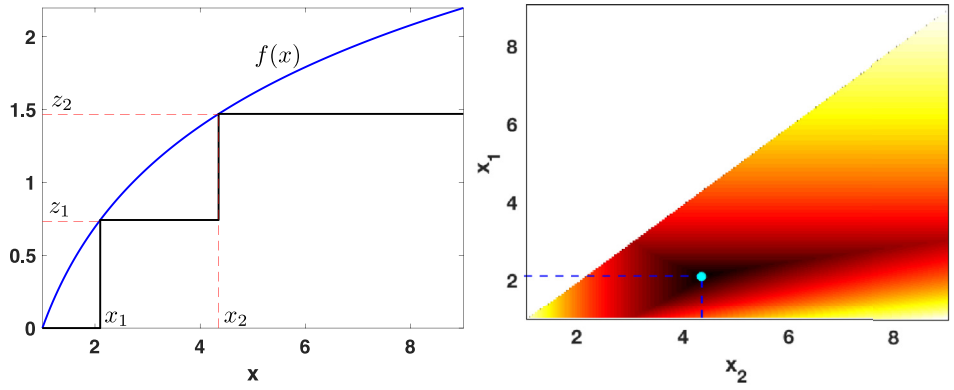
Let us consider the L_∞ distance (i.e., $p = \infty$),

$$\begin{aligned} C_\infty(x_1) &= \max_{x \in [a,b]} |f(x) - \phi_0(x)| \\ &= \max_{x \in [a,b]} [|f(x_1) - f(a)|, |f(x_1) - f(b)|], \end{aligned}$$



(a) (b)

Fig. A.1. **(a)** Optimal piecewise constant approximation $\phi(x)$ with $M = 1$ node. **(b)** The cost function $C_\infty(x_1)$ and its minimum at x_1^*.



(a) (b)

Fig. A.2. **(a)** Optimal piecewise constant approximation $\phi_0(x)$ with $M = 2$ nodes. **(b)** The cost function $C_\infty(x_1, x_2)$ and its minimum at (x_1^*, x_2^*). Note that $C_\infty(x_1, x_2)$ is defined within the simplex such that $x_1 \leq x_2$, by definition; it can be also considered that $C_\infty(x_1, x_2) = C_\infty(x_2, x_1)$.

$$= \max_{x\in[a,b]} [f(x_1) - f(a), f(b) - f(x_1)] \tag{A.4}$$

The problem consists in finding the optimal node x_1^* such that $x_1^* = \arg\min C_\infty(x_1)$. This optimal point x_1^* will then satisfy the condition

$$f(x_1^*) - f(a) = f(b) - f(x_1^*). \tag{A.5}$$

This is because $c_a \equiv |f(x_1) - f(a)|$ will decrease as $c_b \equiv |f(x_1) - f(b)|$ increases, and vice versa, due to the monotonicity of $f(x)$. Since we are taking the max between the two, the lowest value that C_∞ can take is the point where they are equal $c_a = c_b$ (see Fig. A.1), as any divergence from that would lead to one of the terms being higher.

Using Eq. (A.5), since we are assuming that f is monotonic (thus invertible), we can also write

$$f(x_1^*) = \frac{f(b) + f(a)}{2}, \tag{A.6}$$

and, since we have assumed that $f(x)$ is monotonic, thus invertible, we have

$$x_1^* = f^{-1}\left(\frac{f(b) + f(a)}{2}\right). \tag{A.7}$$

Fig. A.1 illustrates the above reasoning. Note that, if $f(x)$ is non-linear, $x_1^* \neq \frac{a+b}{2}$ (as a space filling/Latin hypercube strategy might suggest). The expression of x_1^* is an extended mean, which takes into account information regarding the non-linearity $f(x)$.

A1. Generic number of nodes (M > 1)

Let us assume now that we may place M nodes $x_1, x_2, \ldots, x_M$ in order to achieve an optimal emulator of f with respect to the C_∞ norm. The optimal nodes $x_1^*, x_2^*, \ldots, x_M^*$ will then satisfy the condition

$$\begin{aligned} f(x_1^*) - f(a) &= f(x_2^*) - f(x_1^*) = \ldots f(x_{M-1}^*) - f(x_{M-2}^*) \\ &= f(b) - f(x_{M-1}^*). \end{aligned} \tag{A.8}$$

The point $(x_1^*, x_2^*, \ldots, x_M^*)$ is a minimum for $C_\infty(x_1, x_2, \ldots, x_M)$ and is unique. In order to see this, let us define $d_1 = f(x_1) - f(a)$, $d_2 = f(x_2) - f(x_1), \ldots, d_m = f(x_m) - f(x_{m-1}), \ldots, d_M = f(x_M) - f(x_{M-1})$, and $d_{M+1} = f(b) - f(x_M)$. With this definition we reach the minimum for $C_\infty(x_1, x_2, \ldots, x_M)$ when all distances $\{d_m\}_{m=1}^M$ are equal to $f(b) - f(a)$ divided by $M + 1$:

$$d_1^* = d_2^* = \ldots = d_{M+1}^* = \frac{f(b) - f(a)}{M+1} \equiv c_{\text{MIN}} \tag{A.9}$$

This can be seen from the fact that the distances satisfy $d_m \geq 0$ for $m = 1, 2, \ldots, M$ due to the monotonicity of f, and they sum to a constant

$$\sum_{m=1}^{M+1} d_m = f(b) - f(a) \tag{A.10}$$

Thus if one d_m decreases, one or all other have to increase. This implies that any configuration of $x_1, x_2, \ldots, x_M$ resulting in $d_j <$

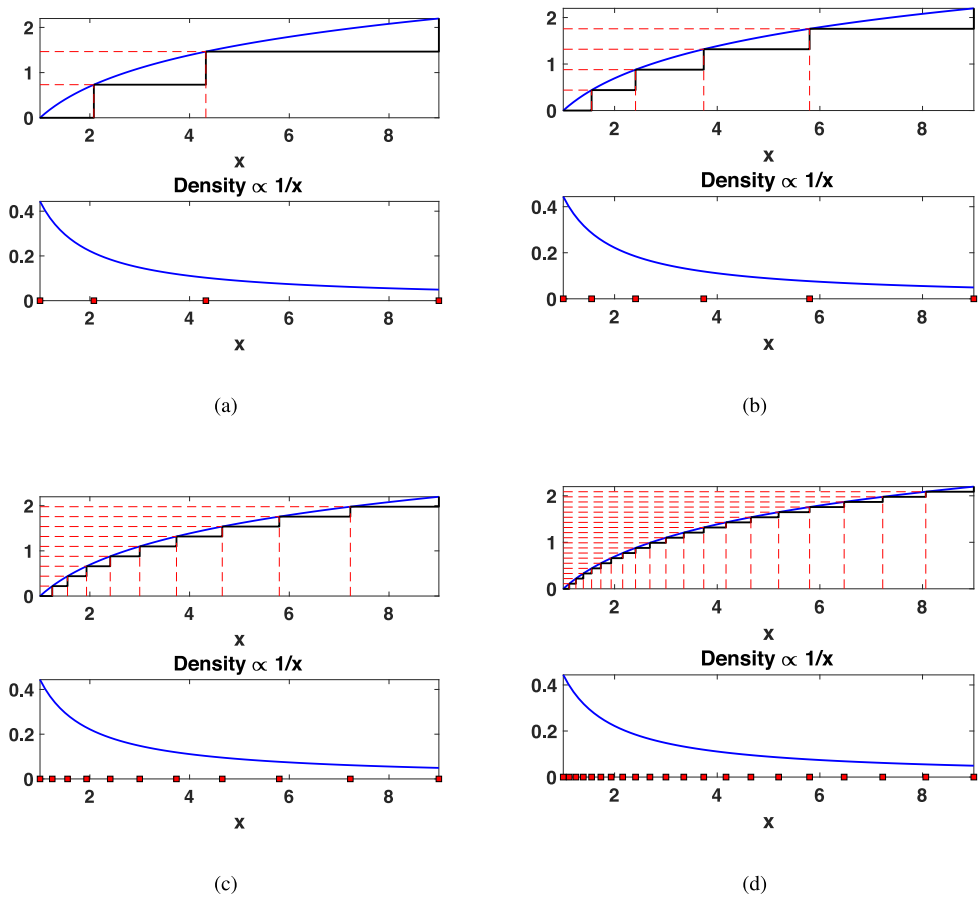


Fig. A.3. Illustration of function and optimal location of nodes (top) and density proportional to gradient (bottom). For (a)-(d) the number of nodes are 2, 4, 10 and 20 respectively.

$c_{\texttt{MIN}}$ for some j, will lead $d_k > c_{\texttt{MIN}}$ for one or more k. Therefore, the configuration of corresponding to (A.9) is optimal.

Following the above logic, the optimal locations of M nodes, $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, can be obtained by:

1. Dividing with a uniform grid formed by M points the interval $[f(a), f(b)]$ (image of $[a, b]$),
$$y_m = f(a) + m\frac{f(b) - f(a)}{M+1}, \qquad m = 1, \ldots, M, T \tag{A.11}$$
2. Finding the x_m such that $f(x_m) = y_m$, i.e., since we assume that $f(x)$ is invertible,
$$x_m = f^{-1}(y_m), \qquad m = 1, \ldots, M. \tag{A.12}$$
See Fig. A.2 for an example with $M = 2$ nodes.

A2. Distributions of nodes

We have seen that the auxiliary points y_m are obtained using a uniform grid in the interval $[f(a), f(b)]$ (image of $[a, b]$). Therefore, $y_1, \ldots, y_M$ is a quasi-Monte Carlo sequence distributed uniformly in $[f(a), f(b)]$, i.e.,

$$y_m \sim \mathcal{U}([f(a), f(b)]), \qquad m = 1, \ldots, M, \tag{A.13}$$

Following Eq. (A.12), we can find the distribution of the nodes x_m since they are obtained by transforming the points y_m through the function $f^{-1}(\cdot)$. Hence, following the expression of the transformation of a random variable, we have

$$x_m \sim p_X(x) = p_Y(f(x))\left|\frac{df}{dx}\right| \tag{A.14}$$

$$\propto \left|\frac{df}{dx}\right|, \qquad m = 1, \ldots, M, \tag{A.15}$$

Therefore, the set of nodes $x_1, \ldots, x_M$ is a quasi-Monte Carlo sequence with density $p_X(x) \propto \left|\frac{df}{dx}\right|$ and if f is increasing, we can write $p_X(x) \propto \frac{df}{dx}$. See Fig. A.3 for an illustration of this. For higher input dimension than 1 we have

$$\mathbf{x}_m \sim p_X(\mathbf{x}) \propto |\nabla f(\mathbf{x})|. \tag{A.16}$$

References

[1] T. Santner, B. Williams, W. Notz, The Design and Analysis of Computer experiments, Springer Verlag, 2003.
[2] B. Wescott, Every Computer Performance Book: How to Avoid and Solve Performance Problems on The Computers You Work With, first ed., CreateSpace Independent Publishing Platform, USA, 2013.
[3] D. Gorissen, L. De Tommasi, K. Crombecq, T. Dhaene, Sequential modeling of a low noise amplifier with neural networks and active learning, Neural Comput. Appl. 18 (5) (2009) 485–494.
[4] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, K. Crombecq, A surrogate modeling and adaptive sampling toolbox for computer based design, J. Mach. Learn. Res. 11 (Jul) (2010) 2051–2055.
[5] M. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc. Ser. B 63 (3) (2001) 425–450.
[6] A. O'Hagan, Bayesian analysis of computer code outputs: a tutorial, Reliab. Eng. Syst. Saf. 91 (10–11) (2006) 1290–1300.
[7] J. Oakley, Bayesian Uncertainty Analysis for Complex Computer Codes, University of Sheffield, 1999 Ph.D. thesis.
[8] Z. Wang, S. Yan, C. Zhang, Active learning with adaptive regularization, Pattern Recognit. 44 (10) (2011) 2375–2383.
[9] Y. Yang, M. Loog, A benchmark and comparison of active learning for logistic regression, Pattern Recognit. 83 (2018) 401–415.
[10] M.C. Shewry, H.P. Wynn, Maximum entropy sampling, J. Appl. Stat. 14 (2) (1987) 165–170.

[11] M.E. Johnson, L.M. Moore, D. Ylvisaker, Minimax and maximin distance designs, J. Stat. Plan. Inference 26 (2) (1990) 131–148.
[12] J. Sacks, S.B. Schiller, W.J. Welch, Designs for computer experiments, Technometrics 31 (1) (1989) 41–47.
[13] J.L. Loeppky, L.M. Moore, B.J. Williams, Batch sequential designs for computer experiments, J. Stat. Plan. Inference 140 (6) (2010) 1452–1464.
[14] C.M. Ryan, C.C. Drovandi, A.N. Pettitt, Optimal Bayesian experimental design for models with intractable likelihoods using indirect inference applied to biological process models, Bayesian Anal. 11 (3) (2016) 857–883.
[15] C.S. Gillespie, R.J. Boys, Efficient construction of Bayes optimal designs for stochastic process models, Stat. Comput. (2019) 1–10.
[16] P. Mitra, B.U. Shankar, S.K. Pal, Segmentation of multispectral remote sensing images using active support vector machines, Pattern Recognit. Lett. 25 (9) (2004) 1067–1074.
[17] D. Tuia, J. Muñoz-Marí, G. Camps-Valls, Remote sensing image segmentation by active queries, Pattern Recognit. 45 (6) (2012) 2180–2192.
[18] S. Seo, M. Wallat, T. Graepel, K. Obermayer, Gaussian process regression: active data selection and test point rejection, in: Mustererkennung 2000, Springer, 2000, pp. 27–34.
[19] H. Wang, J. Li, Adaptive Gaussian process approximation for Bayesian inference with expensive likelihood functions, Neural Comput. 30 (11) (2018) 3072–3094.
[20] A. Ajdari, H. Mahlooji, An adaptive exploration-exploitation algorithm for constructing metamodels in random simulation using a novel sequential experimental design, Commun. Stat.-Simul.Comput. 43 (5) (2014) 947–968.
[21] D. Busby, Hierarchical adaptive experimental design for gaussian process emulators, Reliab. Eng. Syst. Saf. 94 (7) (2009) 1183–1193.
[22] D. Wu, C.-T. Lin, J. Huang, Active learning for regression using greedy sampling, Inf. Sci. 474 (2019) 90–105.
[23] A. Krause, A. Singh, C. Guestrin, Near-optimal sensor placements in gaussian processes: theory, efficient algorithms and empirical studies, J. Mach. Learn. Res. 9 (Feb) (2008) 235–284.
[24] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in: Advances in Neural Information Processing Systems, 2012, pp. 2951–2959.
[25] S. Marmin, D. Ginsbourger, J. Baccou, J. Liandrat, Warped gaussian processes and derivative-based sequential designs for functions with heterogeneous variations, SIAM/ASA J. Uncertainty Quantif. 6 (3) (2018) 991–1018.
[26] D.H. Svendsen, L. Martino, J. Vicent, G. Camps-Valls, Multioutput automatic emulator for radiative transfer models, in: IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2018, pp. 4019–4022.
[27] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, New York, 2006.
[28] G. Camps-Valls, J. Verrelst, J. Muñoz Mari, V. Laparra, F. Mateo-Jiménez, J. Gomez-Dans, A survey on gaussian processes for earth observation data analysis, IEEE Geosci. Remote Sens. Mag. 2 (6) (2016).
[29] M.A. Alvarez, D. Luengo, M.K. Titsias, N.D. Lawrence, Efficient multioutput gaussian processes through variational inducing kernels, in: International Conference on Artificial Intelligence and Statistics, 2010, pp. 25–32.
[30] M.A. Alvarez, L. Rosasco, N.D. Lawrence, Kernels for vector-valued functions: a review, Found. Trends Mach. Learn. 4 (3) (2012) 195.
[31] R.K.S. Hankin, et al., Introducing multivator: a multivariate emulator, J. Stat. Softw. 46 (8) (2012) 1–20.
[32] M. Alvarez, D. Luengo, N. Lawrence, Linear latent force models using gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2693–2705.
[33] A.G. Wilson, D.A. Knowles, Z. Ghahramani, Gaussian process regression networks, in: J. Langford, J. Pineau (Eds.), Proceedings of the 29th International Conference on Machine Learning (ICML), Omnipress, Edinburgh, 2012.
[34] D. Luengo-Garcia, M. Campos-Taberner, G. Camps-Valls, Latent force models for earth observation time series prediction, 2016 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2016), Salerno, Italy, 2016.
[35] L. Martino, V. Elvira, D. Luengo, J. Corander, F. Louzada, Orthogonal parallel MCMC methods for sampling and optimization, Digit. Signal Process. 58 (2016) 64–84.
[36] S.K. Kirkpatrick, C.D.G. Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[37] P. Bratley, B. Fox, Algorithm 659: Implementing Sobol's quasirandom sequence generator, ACM Trans. Math. Softw. (TOMS) 14 (1) (1988) 88–100.
[38] M.D. McKay, R.J. Beckman, W.J. Conover, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (2) (1979) 239–245, doi:10.1080/00401706.1979.10489755.
[39] S. Jacquemoud, W. Verhoef, F. Baret, C. Bacour, P. Zarco-Tejada, G. Asner, C. François, S. Ustin, PROSPECT+ SAIL models: a review of use for vegetation characterization, Remote Sens. Environ. 113 (2009) S56–S66.
[40] M. Campos-Taberner, F. García-Haro, G. Camps-Valls, G. Grau-Muedra, F. Nutini, A. Crema, M. Boschetti, Multitemporal and multiresolution leaf area index retrieval for operational local rice crop monitoring, Remote Sens. Environ. 187 (2016) 102–118, doi:10.1016/j.rse.2016.10.009.
[41] M. Weiss, F. Baret, S2ToolBox Level 2 products: LAI, FAPAR, FCOVER, 2016.
[42] P. Ruiz, J. Mateos, G. Camps-Valls, R. Molina, A.K. Katsaggelos, Bayesian active remote sensing image classification, IEEE Trans. Geosci. Remote Sens. 52 (4) (2013) 2186–2196.

Daniel Heestermans Svendsen received a B.Sc. in Physics and Nanotechnology and M.Sc. in Mathematical Modelling and Computation from the Technical University of Denmark, in 2014 and 2016 respectively. He is currently doing his PhD at the Image and Signal Processing (ISP) group at the Universitat de Valncia, where his main research interests are Gaussian Processes and incorporation of physical knowledge in statistical methods.

Luca Martino, was born in Palermo, Italy, 1980. He received his M.Sc. degree in electronic engineering from the Politecnico di Milano, Italy, and obtained his PhD in Statistical Signal Processing from the Universidad Carlos III de Madrid, Spain, in 2011. Later, he spent two years in the Department of Statistics at the University of Helsinki, Finland. He carried out a postdoc research at Sao Paulo Research Foundation FAPESP, and at Universitat de Valncia. Currently, he is assistant professor at Universidad Carlos III de Madrid.

Gustau Camps-Valls received a PhD in Physics in 2002 from the Universitat de Valncia and he is currently Full professor in Electrical Engineering, and coordinator in the Image and Signal Processing (ISP) group, http://isp.uv.es, at the Universitat de València. He is interested in the development of machine learning algorithms for geoscience and remote sensing data analysis.